

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
GÉPÉSZMÉRNÖKI KAR
MŰSZAKI MECHANIKAI TANSZÉK



Estimating the coefficient of friction based on the memory effect of an elastic tire

Györök Bendegúz Máté, BSc 7. félév

Konzulens:

Dr. Takács Dénes
egyetemi docens

BUDAPEST, 2022 november

Contents

Abstract	IV
Kivonat	V
List of figures	VII
List of tables	VIII
1 Introduction	1
1.1 Overview of the literature	2
2 Modeling and simulation of the investigated mechanical system	3
2.1 The brush tire model	4
2.2 Numerical simulations based on the brush model	6
2.3 Results of simulation	7
3 Estimation of the coefficient of friction based on the brush tire model	10
3.1 Analytical approximation for the contact line deformation function	10
3.2 Estimation method for the coefficient of friction	15
4 Estimation of the coefficient of friction using machine learning	19
4.1 Estimating with neural networks	19
4.1.1 A brief introduction to artificial neural networks	19
4.1.2 Estimating the coefficient of friction using neural networks	22
4.2 Estimation process implementing a Scientific Machine Learning (SciML) model	24
4.2.1 Universal differential equations within scientific machine learning	25
4.2.2 Using scientific machine learning to estimate the coefficient of friction	26
5 Measurement	28
5.1 Measurement set-up	28
5.2 Identification measurements	29
5.2.1 Identifying the coefficient of friction for different measurement plates	29
5.2.2 Identifying the lateral stiffness of the tire	31

5.3	Evaluating the proposed calculation methods using the measured data-sets	33
5.3.1	Estimation based on the analytic modeling approach	34
5.3.2	Estimation with machine learning based methods	36
6	Summary and conclusion	38
6.1	Results	38
6.2	Outlook, further development	40
	Reference	44

Abstract

For precise vehicle motion control, accurate information about the tire-ground relationship is essential. However, the friction properties are changing with time, therefore the identification process is highly complex [1]. The self-aligning torque induced in the contact patch of a steered tire is in close connection with the coefficient of friction. This phenomenon offers a possibility to estimate the friction properties based on the measured steering torque. The goal of this report is to analyze the dynamic effects in the contact region of an elastic tire and the ground.

The so-called brush tire model [2] was selected for the investigation, which is a widely used, simplified analytical model that explains the forces caused by the tire deformation. Based on the brush model, a calculation method to estimate the coefficient of friction is derived in the study. The memory effect of the tire-ground relation is taken into consideration during the derivation, as a result, the lateral deformations can be expressed with a simplified manner [3].

Beside the elaborated estimation procedure, the use of machine learning methods was also investigated. Nowadays, machine learning methods are widely used, for example, the recommendation algorithms of social media companies, as well as control systems of autonomous vehicles can be based on these techniques. In contrast to the previously mentioned applications, in this study, a simpler neural network was trained to estimate the coefficient of friction.

Along with the more “classical” machine learning methods, a scientific machine learning (SciML [4]) process was also implemented, which is a relatively new research area. The method is based on data-driven machine learning as well as model-based mechanical formulas and equations, in such way that the accuracy of the estimation can be enhanced. In this work, the nonlinear equation for the coefficient of friction was complemented with a neural network-based learning algorithm. The SciML method were tested, whether it can compensate the errors resulting from the unmodeled dynamics, the inaccuracy of the numerical computations and the measurement noise.

The results of the previously presented estimation methods were compared based on different aspects: accuracy, run time, robustness, extrapolating properties, etc. The methods were tested with simulations as well as measurements. The measured datasets were gathered with an experiment set-up manufactured for this purpose.

Kivonat

A járművek precíz mozgásszabályozásához elengedhetetlen a kerék-talaj kapcsolat súrlódási viszonyainak ismerete. A súrlódási viszonyok azonban időben nem állandóak a mozgás során, így azok azonosítása bonyolult feladat [1]. Mivel a kormányzott gumikerék érintkezési tartományában fellépő önbeálló nyomaték szoros kapcsolatban van a kerék-talaj kapcsolatot jellemző súrlódási tényezőkkel, így kézenfekvő lehetőségként adódik, hogy a kormányzási nyomaték mérése által következtessünk a súrlódási együttható értékeire. A dolgozat a kerék-talaj kapcsolatban lejátszódó dinamikai folyamatok elemzését tűzi ki célul.

A vizsgálatainkhoz a gumikerék ún. kefe-modelljét alkalmaztuk [2], amely egy elterjedt egyszerűsített, analitikusan kezelhető bonyolultságú modell a gumikerék-talaj kapcsolat leírására. A kefe-modell alapján a dolgozatban kidolgozásra kerül egy számítási eljárás a súrlódási együttható becslésére. Az eljárásban felhasználtuk a kerék-talaj kapcsolatot jellemző memória-hatást, melynek segítségével a gumikerék laterális deformációja egyszerűsített módon írható le [3]. A kidolgozott becslő eljárás alkalmazása mellett a jelenleg nagy népszerűségnek örvendő gépi tanulási módszerek használhatóságát is vizsgáltuk. A gépi tanulási módszerek rendkívül sok területen megjelennek az életünkben, például a nagy szociális média felületek ajánló algoritmusaiiban vagy az autonóm járművek szabályozásaiban találkozhatunk velük. Az imént említett példák összetett algoritmusaihoz képest a dolgozatban egy egyszerű neurális hálózatot betanítva teszteltük, hogy mennyire alkalmas a súrlódási együttható becslésére.

A klasszikus gépi tanulási algoritmusok mellett külön figyelmet fordítottunk a tudományos gépi tanulás (scientific machine learning, SciML [4]) viszonylag új kutatási területre. A módszer alapja, hogy a mérési adatokon alapuló gépi tanulás mellett, ismert, tudományosan megalapozott mechanikai összefüggéseket, egyenleteket is figyelembe véve az algoritmus, így növelve a becslés pontosságát. A dolgozatban a súrlódási együtthatóra felírt nemlineáris egyenletet kiegészítve gépi tanulással, azt teszteltük, hogy a nem modellezett jelenségekből, a numerikus eljárások pontatlanságából és a mérési zajból adódó hibákat sikerül-e a SciML módszerrel kompenzálni.

A dolgozatban a fentebb ismertetett becslő eljárások eredményeit hasonlítjuk össze különböző szempontok alapján (pontosság, futási idő, robusztusság, extrapolálás, stb.). Az eljárásokat szimulált adatsoron, valamint erre a célra készített kísérleti berendezésen végzett mérésekkel is teszteltük.

List of Figures

2.1	Axonometric and side view of the investigated mechanical system.	3
2.2	Schematic view of the tire brush model.	4
2.3	In-plane model of the brush model.	5
2.4	Illustrating the mechanical system for deriving the equation for the kinematic constraint of sticking.	6
2.5	Simulation results.	8
2.6	The deformation function of the center-line of the tire, illustrated at the given time instants.	9
3.1	Deformation function in case of three distinct deformation zones and positive yaw angle.	11
3.2	Analytical deformation functions compared to results obtained by numerical simulation.	12
3.3	In plane model showing five deformation zones.	13
3.4	Comparing the analytical deformation function with the results obtained by numerical simulation.	14
3.5	State machine representation of the approximation process for the deformation function.	15
3.6	Estimating the yaw angle, with the value of the coefficient of friction given to the algorithm.	16
3.7	Estimation results for the coefficient of friction using simulated data sets. .	17
3.8	Illustrating $f(\mu)$ at two different time instances.	18
3.9	The results of the estimation showing the causes of inaccuracies.	18
4.1	Representation of the layers in neural networks.	20
4.2	Different activation functions illustrated.	21
4.3	A single neuron illustrated with three inputs.	21
4.4	Training inputs of the neural network.	22
4.5	Error of the models during a typical training process.	23
4.6	Results of the estimation with the simple neural network model.	24
4.7	Comparing the results of the estimation with neural network with the results from analytical method.	25
4.8	The results of the coefficient of friction estimation with the proposed SciML method on simulated data.	27

5.1	Experimental rig with the used devices.	28
5.2	Experiment set-up for measuring the coefficients of friction	30
5.3	Results of the coefficient of friction identification measurement for one of the emery papers.	30
5.4	Five measurement signals of the lateral stiffness identification experiment for one of the contact patch lengths.	31
5.5	Identification results for the lateral stiffness and the torsional damping. . .	33
5.6	The yaw angle signal and the determined $\psi_{\max}$ values of an arbitrary ex- periment case.	34
5.7	The fragmentation number and the direction of angular motion for an ar- bitrary measurement case.	35
5.8	Results of the estimation for one of the measurement cases.	36
5.9	Box chart illustration of the estimation error and the time cost with the defined dimensionless time cost quantity.	37
5.10	Error of the neural network model during training on the measurement data. 38	
5.11	Results of the estimation methods for one of the validation measurements. 39	
5.12	Comparing the accuracy of the estimations for the validation measurement cases, as well as the time cost of the computations.	40

List of Tables

2.1	List of parameters used in the numerical simulations.	8
3.1	List of parameters used in the numerical simulations for testing the developed estimation algorithm.	16
5.1	Identified coefficients of friction.	31
5.2	Identified lateral stiffness and torsional damping.	33
6.1	Summarized results of each estimation method.	40

1 Introduction

For precise vehicle motion control, accurate information about the tire-ground relationship is essential. However, the friction properties are not observable, therefore it requires special methods to estimate the coefficient of friction. In the literature, several methods can be found, but none of them provides an accurate, fast and robust technique, thus the development of an algorithm, that fulfills the above mentioned criterion, is an important topic.

The self-aligning torque induced in the contact patch of a steered tire is in close connection with the coefficient of friction. This phenomenon offers a possibility to estimate the friction properties based on the measured steering torque. The aim of this work is to present three methods for estimating the coefficient of friction between the road surface and the tire based on analyzing the dynamic effects in the contact region. The first method solely relies on the analytic modeling of the tire-ground contact. The second method is based on a machine learning technique, neglecting the physics-based part of the system. And finally, the third method combines the advantages of the previous two with a relatively new method in the fields of machine learning.

While physics based modeling is known to provide reasonably accurate solutions for centuries, the field of data-driven modeling with machine learning was started to be applied in the last 40-50 years. Although we are able to find the appropriate equations that describe the dynamics of elastic tires, the modeling is based on assumptions and neglects, which cause errors. In contrast, machine learning techniques often provide accurate solutions, but the inner structure of the network is hard to see through, the results are hard to interpret on a physical level. Scientific machine learning (SciML) combines the physics-based modeling with data-driven approximations, therefore provides more accurate results, because the unmodeled effects are compensated.

To investigate the dynamic effects in the contact region of an elastic tire and the road surface, this study has the following structure. In Section 2, the investigated mechanical system, the so-called towed wheel model is introduced, then an analytic tire model - the brush tire model - is being discussed. A simulation environment is developed based on the brush model, in order to analyze the dynamics of the system.

In Section 3, an estimation method is derived, based on the analytic brush tire model. First, the deformation of the center line of an elastic tire is expressed with an analytic function, then the estimation process is presented with the results on simulated data sets.

Section 4 presents the other two estimation methods. After a brief introduction to artificial neural networks, an estimation technique is discussed using neural networks. Then, a relatively new topic is discussed, namely the field of scientific machine learning, which effectively combines data driven modeling with physics based equations. Within SciML, a new method is proposed to solve nonlinear dynamical problems. The results of all three estimation algorithm are compared on simulation data.

In Section 5, an experiment process is discussed. An already established measurement set-up at the Department of Applied Mechanics was modified, in order to test the proposed estimation algorithms on measured data. After discussing the experiment process, the results of the estimations are compared based on different aspects.

In Section 6, the achieved results are summarized, and the final conclusions are drawn. The possible practical applications and the options for further development are also discussed.

1.1 Overview of the literature

Accurate estimation of the frictional properties in the contact region of road surface and elastic tires is an essential problem in the field of vehicle dynamics. Knowledge about the coefficient of friction makes the vehicle control systems more robust and accurate. Therefore several studies have been done, in order to develop an estimation method for the friction parameters of tires.

The previously proposed methods can be classified into several groups: vision based, wheel vibration based and vehicle motion based. While vision based predictions have several possibilities in store for the future, with the rapidly improving vision recognition softwares, currently they are not reliable in bad lighting [5]. Tire vibration models monitor the longitudinal velocity of the tires to estimate the tire-road friction [6]. In our work, the tires will be investigated with no longitudinal motion, therefore these methods could not be applied. Apart from that, this method suffers from high frequency noises. And the last group, that monitors the vehicle dynamics for estimating the friction, usually relies on GPS signals of the car [7], which was not realizable for us, and also would not provide accurate results with no longitudinal motion.

Therefore it was necessary to develop an estimation process, that is able to provide approximations for the coefficient of friction based on only monitoring the lateral dynamics of the elastic tires.

2 Modeling and simulation of the investigated mechanical system

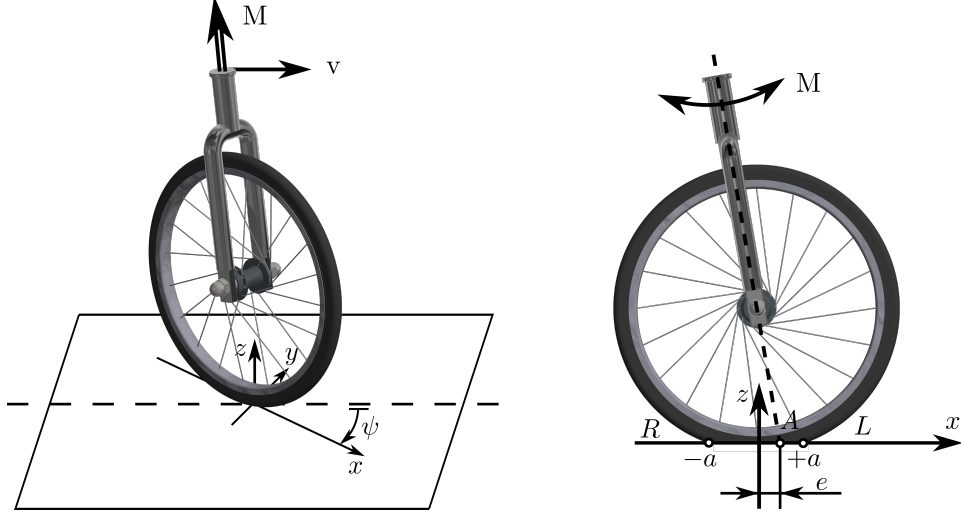


Figure 2.1: Axonometric (left) and side (right) view of the caster-wheel system.

To investigate the tire-ground contact, we use the so-called towed wheel model. In Figure 2.1, the mechanical system is illustrated with the coordinate system and some of the geometrical parameters. In order to describe the tire deformation, the (x, y, z) coordinate system is used. Its origin is fixed to the center of the tire-ground contact patch, while the x and y axes are aligned to the longitudinal and lateral directions of the wheel, respectively. The z -axis is normal to the plane of the ground.

In the side view of the mechanical system (right panel of Figure 2.1), the most relevant geometrical parameters regarding the elastic tire are highlighted. The length of the contact patch is denoted by $2a$. The end points of the contact region, where $x = a$ are called the leading edge (L), the $x = -a$ points are called rear edge (R). The system is actuated with the steering torque M as shown in Figure 2.1.

In the right panel of Figure 2.1, the axis of revolution is illustrated with a dotted line. The point, where this line crosses the x -axis, is defined as point A . The distance between the center point of the contact patch and the point A is called as caster length (or trail) and denoted by e . The mass moment of inertia of the steered caster-wheel system with respect to king pin axis is considered.

In this work, only the quasi-static case is modeled, where the longitudinal velocity of the system is zero ($\mathbf{v} = \mathbf{0}$) and the rotational motion of the wheel is described by steering angle (or yaw angle) $\psi(t)$, which motion induces the deformation of the tire.

2.1 The brush tire model

There are different approaches to model the tire-road relationship. Starting from the 1940's, the vehicle industry showed more and more interest in tire modeling, therefore the first mechanical models were introduced in that time period [1]. Later on, researchers focused more on finding empirical formulas that describe the tire forces. Nowadays, with the advanced computation efforts, combined tire models are being introduced, using both physical models and empirical formulas, moreover finite element methods are used to validate these modern theories.

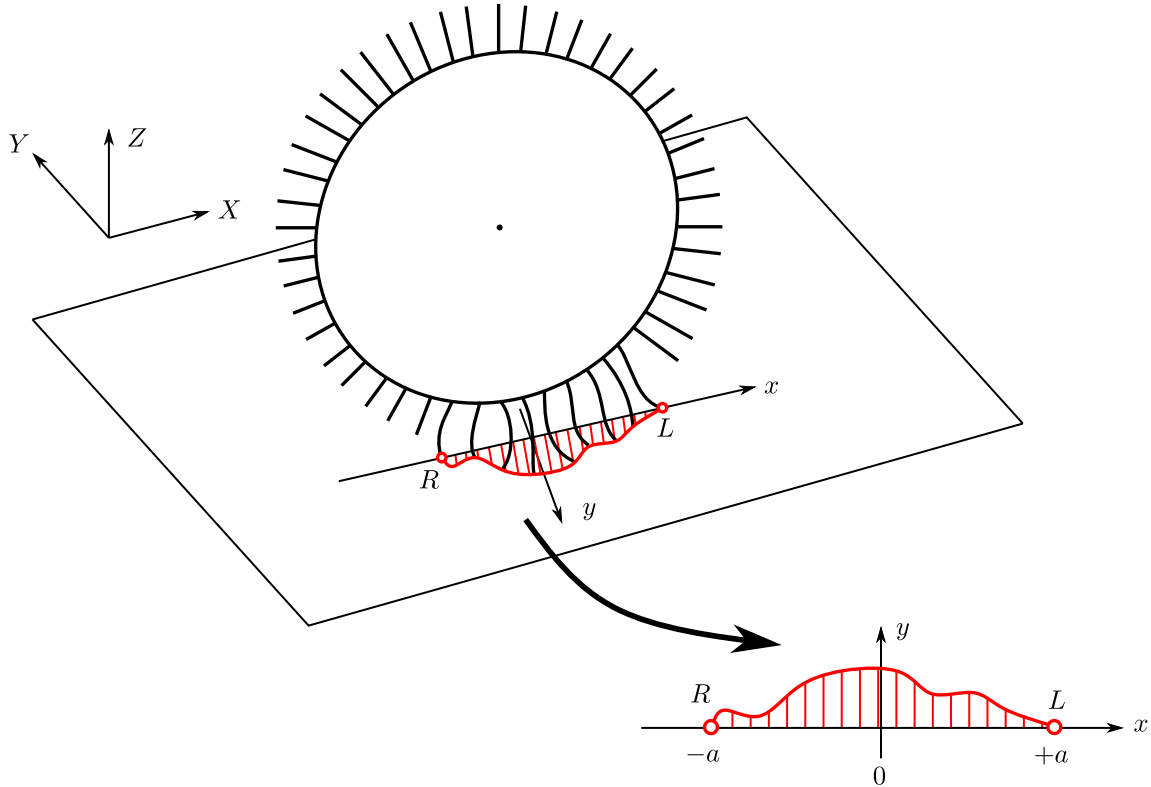


Figure 2.2: Schematic view of the brush tire model. The leading edge is noted by L , the rear edge by R . The in-plane deformation within the contact patch is also illustrated.

The brush tire model [2] is a popular analytical approach to model torques and forces resulting from the deformation of tires. The schematic view of the model can be seen in Figure 2.2. The model consists of tread elements (also called bristles) connected to the carcass. The carcass is represented as a rigid disk. The tread elements are modeled as linear springs and their deformation are independent from each other, therefore outside of the contact patch, zero deformation can be assumed.

An in-plane model of the the brush model can be defined, where the deformation of the elastic bristles are calculated in the $x - y$ plane, as shown in Figure 2.2 and 2.3. Inside the contact patch, the deformation is limited by the available friction between the treads and the road surface. Assuming parabolic normal force distribution (which assumption was proven by measurements in [8]) in the contact region, the limit of the deflection (q_{cr})

is also parabolic

$$q_{cr}^{\pm} = \pm \frac{3F_N\mu}{4a^3k} (a^2 - x^2), \quad (2.1)$$

where F_N is the normal force, μ is the coefficient of friction and k [N/m²] is the distributed lateral stiffness of the elastic bristles (it is assumed to be homogeneous around the carcass).

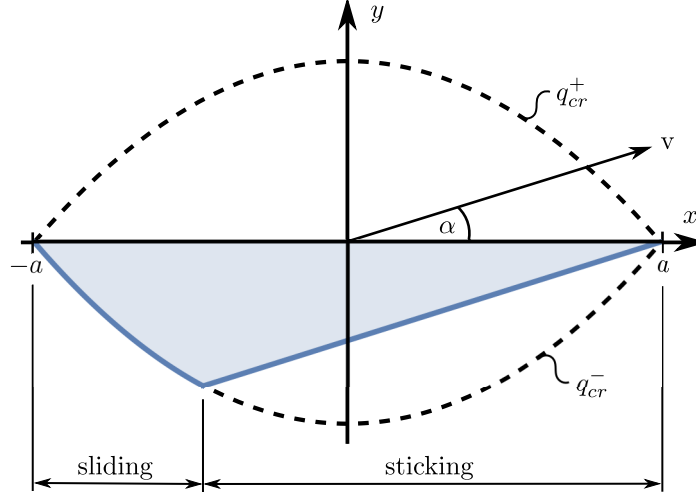


Figure 2.3: In-plane model of the tire brush model, in case of non-zero longitudinal velocity.

Originally, the brush tire model was developed to model tire forces in case of non-zero longitudinal vehicle velocity. In this work, the concept is used to model tire deformations in a quasi-static case, where the longitudinal speed is assumed to be zero. In Figure 2.3, the deformation of an elastic tire is illustrated by the original idea with non-zero longitudinal speed. The sliding and sticking regions within the contact patch are also shown in Figure 2.3. Tire elements enter the contact patch at the leading edge L with zero deformation. In quasi-stationary case, the brush elements start to be deformed according to the side slip angle α , which measures the angle of the wheel center point velocity relative to the wheel center plane. When the deformation of an element reaches the limit of deflection, it enters the sliding zone, and their deformation is limited by the above defined q_{cr} . In the sticking region, the deformation of the bristles can be described as: $q_r(x) = \tan \alpha \cdot (x - a)$. It is important to emphasize that q_r is only valid in this particular case, later on the accurate deformation functions will be derived. In an arbitrary position x , the deformation of the tread elements (q) can be expressed as

$$q(x) = \begin{cases} q_{cr}^-, & \text{when sliding,} \\ q_r, & \text{when sticking.} \end{cases} \quad (2.2)$$

The deformation of the elements cause a torque, namely the self-aligning torque, that acts against the steering torque. Assuming the deformation function is known for $t \in [t_0, t_{\text{end}}]$, the self aligning torque can be expressed as

$$M_z(t) = \int_{-a}^a k \cdot q(x, t) \cdot (x - e) dx. \quad (2.3)$$

2.2 Numerical simulations based on the brush model

For the investigation of the tire-ground relation, a simulation environment was set up based on the brush model. Based on Figure 2.4, let us consider an arbitrary point P in the sticking region. The wheel is towed by the rigid caster with constant (zero) velocity in the X direction. The lateral deformation of the tire can be described with $q(x, t)$ function. It is known that kinematic constraint of sticking requires the absolute velocity to be zero, therefore $\mathbf{v}_P = \mathbf{0}$.

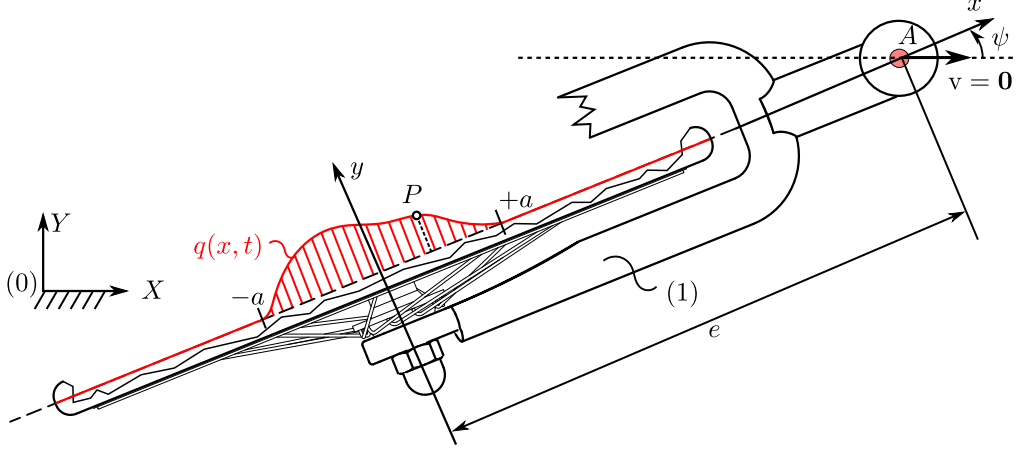


Figure 2.4: Illustrating the mechanical system for deriving the equation for the kinematic constraint of sticking.

Using the notations of relative kinematics

$$\mathbf{v}_{P20} = \mathbf{v}_{P10} + \mathbf{v}_{P21}, \quad (2.4)$$

where $\mathbf{v}_{P20}$ is the absolute velocity of point P , $\mathbf{v}_{P10}$ is the carrier velocity and $\mathbf{v}_{P21}$ is the relative velocity.

The absolute velocity of the carcass, or in other words, the carrier velocity of the tire, can be expressed as

$$\mathbf{v}_{P10} = \mathbf{v}_{A10} + \omega_{10} \times \mathbf{r}_{AP}. \quad (2.5)$$

Since the towing speed is zero in our study, therefore $\mathbf{v}_{A10} = \mathbf{0}$. As a result, $\mathbf{v}_{P10}$ can be expressed in the (x, y, z) coordinate system as

$$\mathbf{v}_{P10} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} -(e-x) \\ q(x, t) \\ 0 \end{bmatrix} = \begin{bmatrix} -q(x, t) \cdot \dot{\psi} \\ -(e-x) \cdot \dot{\psi} \\ 0 \end{bmatrix} \quad (2.6)$$

The relative velocity between the tire and the carcass can be attained by the time derivation of the relative position vector $\mathbf{r}_{P21}$, as

$$\frac{d}{dt} \mathbf{r}_{P21} = \frac{d}{dt} \begin{bmatrix} x \\ q(x, t) \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{q}(x, t) + q'(x, t) \cdot \dot{x} \\ 0 \end{bmatrix}, \quad (2.7)$$

where $\dot{\square}$ notes the partial derivative with respect to time and $\square'$ notes the partial derivative with respect to coordinate x . Thus the time derivative of $q(x, t)$:

$$\frac{d}{dt}q(x, t) = \underbrace{\frac{\partial}{\partial t}q(x, t)}_{\dot{q}(x, t)} + \underbrace{\frac{\partial q(x, t)}{\partial x}}_{q'(x, t)} \cdot \underbrace{\frac{\partial x}{\partial t}}_{\dot{x}}. \quad (2.8)$$

From (2.7), the kinematic constraint equations of sticking can be expressed as

$$-q(x, t) \cdot \dot{\psi} + \dot{x} = 0, \quad (2.9)$$

$$-(e - x) \cdot \dot{\psi} + \dot{q}(x, t) + q'(x, t) \cdot \dot{x} = 0, \quad (2.10)$$

where $x \in [-a, a]$, assuming the whole contact region is sticking.

From (2.9), the translational velocity can be determined:

$$\dot{x} = q(x, t) \cdot \dot{\psi}(t). \quad (2.11)$$

From (2.10), the differential equation for the lateral deformations of the elastic tire elements can be expressed as

$$\dot{q}(x, t) = (e - x) \cdot \dot{\psi}(t) - q'(x, t) \cdot q(x, t) \cdot \dot{\psi}(t), \quad (2.12)$$

where the last term is nonlinear. Considering small deformations and motions, it represents a third-order small term that can be neglected in the analysis.

The equation of motion for the caster-wheel system can be expressed using Newton's second law, as

$$J_A \ddot{\psi}(t) = -b_t \dot{\psi}(t) - M_z(t) + M(t), \quad (2.13)$$

where J_A is the mass moment of inertia with respect to point A , b_t is the torsional damping of the kingpin, M_z is the self-aligning torque of the tire and M is the steering torque.

The governing equation of the system consist of the integro-differential equation (2.13), where the self-aligning moment can be calculated by (2.3), while the lateral deformation is described by the partial differential equation (2.12). Moreover, the sliding in the contact patch is also considered since the deformation is maximized within the contact patch based on (2.1). The equations necessary for the numerical simulation can be summarized as

$$J_A \ddot{\psi}(t) = -b_t \dot{\psi}(t) - \int_{-a}^a k \cdot q(x, t) \cdot (x - e) dx + M(t), \quad (2.14)$$

$$\dot{q}(x, t) \approx (e - x) \cdot \dot{\psi}(t), \quad (2.15)$$

$$|q(x, t)| \leq \frac{3F_N \mu}{4a^3 k} (a^2 - x^2), \quad \forall t \in [t_0, t_{\text{end}}]. \quad (2.16)$$

2.3 Results of simulation

For the numerical simulations, the steering torque input have to be defined by the user. The parameters used in the simulations are summarized in Table 2.1.

Parameter	Notation	Value
Caster length (trail)	l	0.01 m
Contact patch half-length	a	0.02 m
Lateral tire stiffness	k	60000 N/m ²
Mass moment of inertia with respect to A	J_A	0.1 kgm ²
Torsional damping	b_t	0.4827 Nms
Vertical tire load	F_N	180 N
Coefficient of friction	μ	0.5

Table 2.1: List of parameters used in the numerical simulations.

The simulation was set up to calculate the deformation of the contact patch at every time-step, and after the simulation procedure, an animation shows the results. An example steering torque input is shown in Figure 2.5, with the time instants, for which the deformation function $q(x, t)$ is illustrated in Figure 2.6.

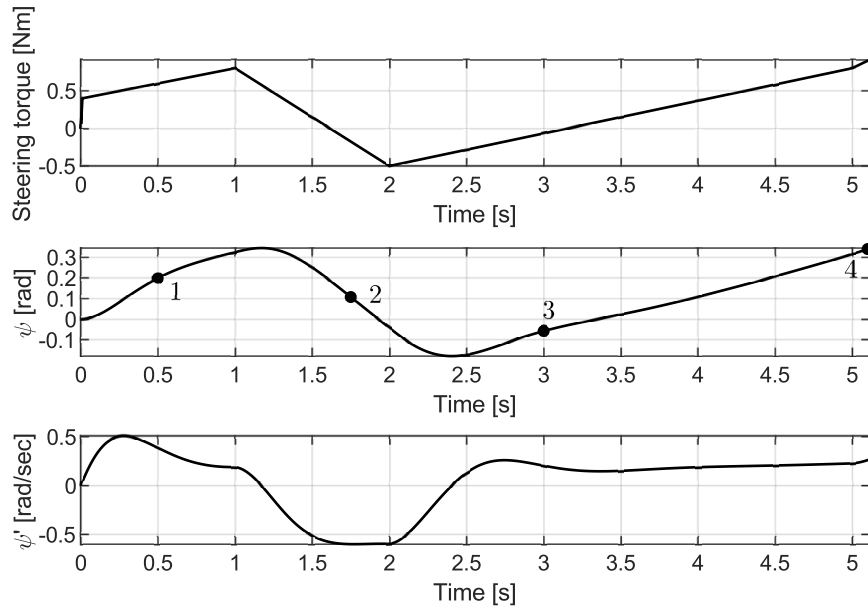


Figure 2.5: Results of the numerical simulation, regarding the yaw angle, angular velocity and steering torque.

It is visible in Figure 2.5, that there was a direction change in the angular motion at about $t = 1.25$ s, and then at $t = 2.4$ s. In Figure 2.6 the first deformation function is showed at $t = 0.5$ s, where three fragmentation zones appear in the contact patch. Then after the first direction change, at $t = 2.75$ s it is shown that, in fact, five zones are present in the contact patch. Between the first and second direction change, the tire was not steered enough to terminate the fourth and fifth zones, therefore after the second change of rotational direction, another two deformation zones are generated, as it is shown at $t = 3$ s. Then the tire is turned continuously in the same direction, hence by the end

of the simulation, only three fragmentation zones remain, as it is illustrated at $t = 5.1$ s.

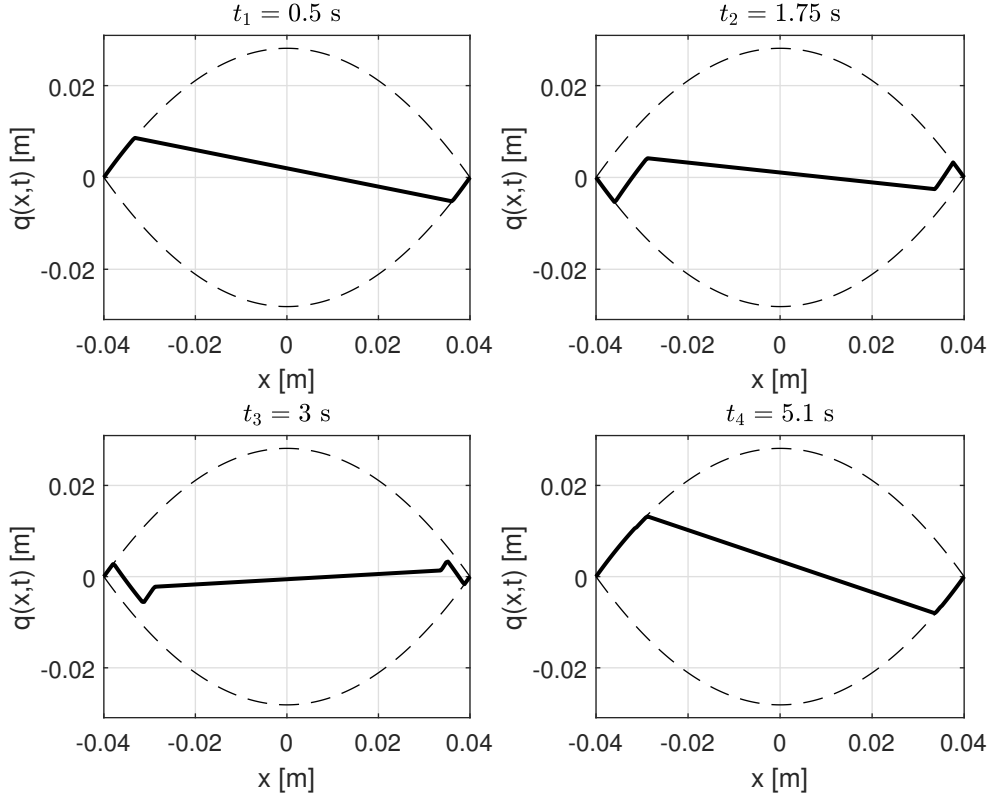


Figure 2.6: The deformation function of the center-line of the tire, illustrated at the given time instants.

The above discussed simulation environment made possible to test the later derived friction estimation methods. The calculation procedure was later implemented into the simulation scripts, therefore it is possible to compare the identified coefficient of friction with the simulation values, within the simulation tool.

3 Estimation of the coefficient of friction based on the brush tire model

Three methods are being derived for estimating the coefficient of friction between the tire and the ground. The first method is based on an analytic tire model - namely the previously introduced brush tire model. The second method is based on a neural network machine learning algorithm, therefore the coefficient of friction is estimated by data-driven modeling, the knowledge about the physical system is neglected. The third algorithm incorporates the principle of the previous two, besides calculating from the physical model of the system, the unmodeled effects are compensated with a machine learning technique.

In the previous section, the brush tire model was introduced in order to model the self-aligning torque resulting from the deformation of the tire. Although the method was originally developed for vehicles with longitudinal motion, the principle of the mechanical model remains the same, when the wheel is standing still and the only actuation is the steering torque.

The principle of the estimation method is based on (2.14). Assuming $\dot{\psi}(t)$, $\ddot{\psi}(t)$, $M(t)$ can be measured and J_A , k , b_t , a , e constants are known, the only unknown in the equation of motion is the deformation function of the contact line. It was previously discussed that the deflection of the elastic elements is dependent on the coefficient of friction between the ground and the tire. Assuming $q(x, t)$ can be expressed as a function of the friction coefficient and known or measured physical properties, μ could be calculated theoretically. In the following chapter, this method will be derived, later on it will be tested with simulated and measured signals.

3.1 Analytical approximation for the contact line deformation function

From (2.14), it is clear that the first task is to express the deformation function with the coefficient of friction. From now on, at $t = 0$ s zero deformation will be assumed and in general, the the angular rotation will begin with a positive direction. From Figure 2.1 it is visible that point A is on the axis of rotation, therefore zero deformation will occur at that point. In the sticking region, the tread elements will remain in the same position in the ground-fixed reference frame, but in the caster-fixed (x, y, z) system they will be revolved with ψ angle around A . And when the deformation reaches q_{cr} , then it will be limited by the available frictional forces.

From Figure 3.1, it is visible, that the deformation function of this type can be expressed

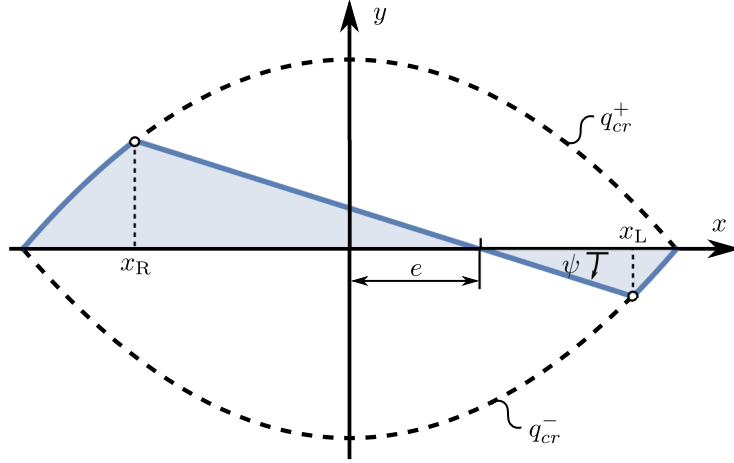


Figure 3.1: In-plane brush model representation of the tire elements deformation. Three distinct zones can be visibly identified, therefore three different equations are needed to describe the deformation function.

as

$$q(x) = \begin{cases} \frac{3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [-a, x_R), \\ -\tan \psi \cdot (x - e), & \text{if } x \in [x_R, x_L), \\ \frac{-3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [x_L, a], \end{cases} \quad (3.1)$$

where x_R and x_L are the points, where the sticking turns into sliding as a result of reaching the critical deformation limit. The values of these variables can be calculated analytically by solving simple algebraic equations.

Equation (3.1) is only valid for deformations, where only three sections are generated by the angular motion and the direction of the revolution is positive at the beginning. For negative directions with three deformation zones, a similar expression can be derived. For the sticking region the deformation can be characterized with the same function as before, but it will reach the deformation limits at different points, therefore

$$q(x) = \begin{cases} \frac{-3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [-a, x_R), \\ -\tan \psi \cdot (x - e), & \text{if } x \in [x_R, x_L), \\ \frac{3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [x_L, a] \end{cases} \quad (3.2)$$

deformation function can be derived. Thus, in general, when three deformation zones are expected, the analytical formula for the contact line deformation can be expressed as

$$q(x) = \begin{cases} \operatorname{sgn}(\psi) \frac{3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [-a, x_R), \\ -\tan \psi \cdot (x - e), & \text{if } x \in [x_R, x_L), \\ -\operatorname{sgn}(\psi) \frac{3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [x_L, a], \end{cases} \quad (3.3)$$

where sgn notes the signum function.

With the numerical simulation, that was introduced in section 2.2, it is possible to test, whether the above described formulas are correct or not. In our analysis, we consider that both the deformations and the angle ψ are small, hence the sticking part can be linearized:

$$q(x) = \begin{cases} \operatorname{sgn}(\psi) \frac{3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [-a, x_R), \\ -\psi \cdot (x - e), & \text{if } x \in [x_R, x_L), \\ -\operatorname{sgn}(\psi) \frac{3 F_N \mu}{4 a^3 k} \cdot (a^2 - x^2), & \text{if } x \in [x_L, a]. \end{cases} \quad (3.4)$$

The linearization caused small errors compared to the simplifications made during the model creation, therefore it was decided to calculate with the linearized equations from now on. In Figure 3.2, it is shown that the previously derived analytical functions are able to accurately describe the deflection of the contact line, if the deformation function has only three zones.

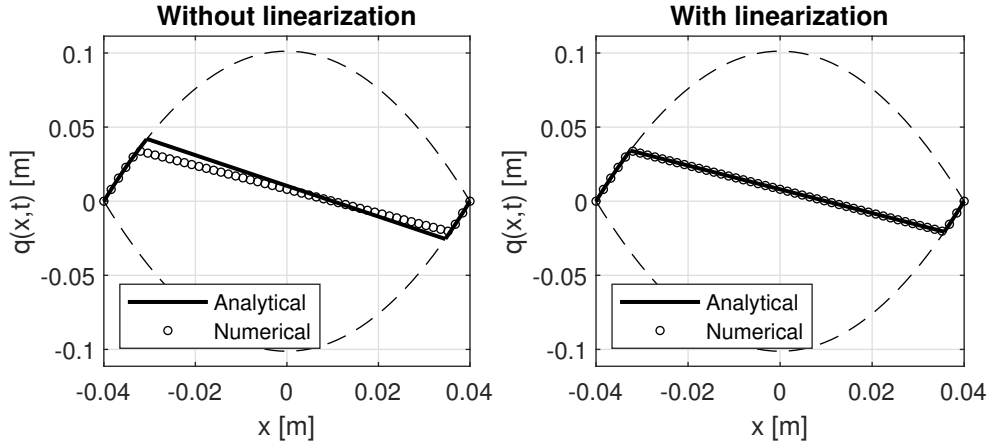


Figure 3.2: Analytical deformation functions compared to results obtained by numerical simulation at a given time-step. On the left the nonlinear form can be seen. On the right the results after linearization are visible.

From Figure 2.6, it can be seen, that more than three deflection zones can occur in the contact line of an elastic tire. For example, when the direction of rotation is changed, the thread elements, that are in the sliding region can revolve back into the sticking region. As a result, in the deformation function parabolic sections appear, which are revolved around point A . An example of a deformation function with such zones can be found in Figure 3.3.

Let us suppose that three deflection zones are present in the contact patch. At a given time-step, the direction of the rotation is changed from positive to negative direction. At that given time, the yaw angle is defined as $\psi_{\max}$, thus the deformation function can be

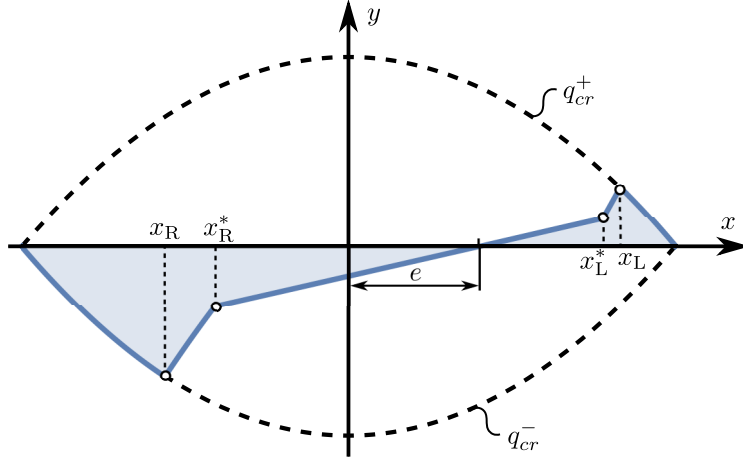


Figure 3.3: In plane model showing five deformation zones. The limits of each section is noted in the figure with the appropriate notation, also used in the equations.

expressed as

$$q(x) = \begin{cases} \frac{-3 F_N \mu}{4 a^3 k} (a^2 - x^2), & \text{if } x \in [-a, x_R], \\ \frac{3 F_N \mu}{4 a^3 k} (a^2 - x^2) + m_R x - m_R e & \text{if } x \in [x_R, x_R^*], \\ -\psi (x - e), & \text{if } x \in [x_R^*, x_L^*], \\ \frac{-3 F_N \mu}{4 a^3 k} (a^2 - x^2) + m_L x - m_L e & \text{if } x \in [x_L^*, x_L], \\ \frac{3 F_N \mu}{4 a^3 k} (a^2 - x^2), & \text{if } x \in [x_L, a], \end{cases} \quad (3.5)$$

where x_R^* and x_L^* are the point of the contact line, where the linear deformation zone crosses the rotated parabolic zones. The rotated parabolas can be calculated as the addition of the parabolic limits of deflection and linear functions that can be parameterized with m_L and m_R . The values of x_R^* and x_L^* can be obtained by calculating x_R and x_L for $\psi_{\max}$ for three deflection zones (equation (3.4)). Then the values of m_R , m_L , x_R and x_L can be obtained by solving simple algebraic equations.

Similarly to Figure 3.2, the above described formulas can be compared to the numerical results. In Figure 3.4 the analytical results are shown next to the numerical deformation values and it is visible that the analytically derived formulas are able to accurately describe the deflections that occur in the contact patch.

The same principle can be used, when the angular direction changes from negative to positive. Three deflection zones turn into five zones, and the deformation function can be

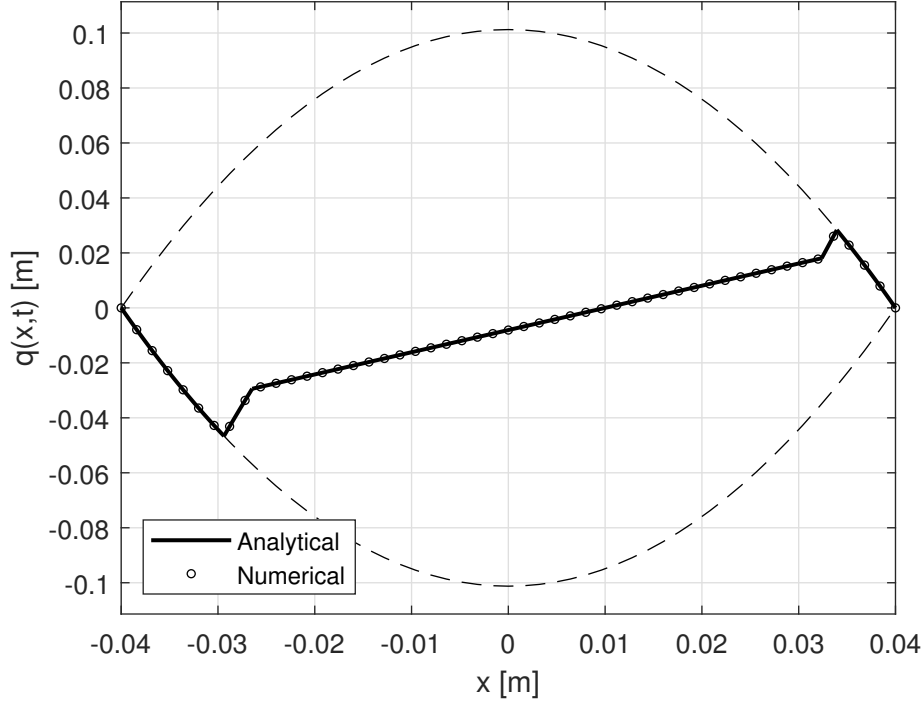


Figure 3.4: Comparing the analytical deformation function with the results obtained by numerical simulation.

expressed as

$$q(x) = \begin{cases} \frac{3 F_N \mu}{4 a^3 k} (a^2 - x^2), & \text{if } x \in [-a, x_R), \\ \frac{-3 F_N \mu}{4 a^3 k} (a^2 - x^2) + m_R x - m_R e, & \text{if } x \in [x_R, x_R^*), \\ -\psi (x - e), & \text{if } x \in [x_R^*, x_L^*), \\ \frac{3 F_N \mu}{4 a^3 k} (a^2 - x^2) + m_L x - m_L e, & \text{if } x \in [x_L^*, x_L), \\ \frac{-3 F_N \mu}{4 a^3 k} (a^2 - x^2), & \text{if } x \in [x_L, a], \end{cases} \quad (3.6)$$

where m_R , m_L , x_R , x_L , x_R^* and x_L^* can be calculated similarly as before.

A similar process can be observed if five deformation zones are present in the contact line and a direction change occurs in the angular motion. As a result more rotated parabolic zones are generated in the sticking region, but the effect of those deformation zones assumed to be negligible, therefore it was not modeled in this work.

Although previously only those cases were discussed, when more and more zones are generated in the deformation function of the contact line, in other words, fragmentation are created with the change of angular direction. The moment when this happens, the yaw angle was defined to be $\psi_{\max}$. When the yaw angle's absolute value exceeds $\psi_{\max}$ ($\psi > -\psi_{\max}$), the fragmented zones reach q_{cr} limit of deflection, therefore the fragmentation disappears. The process of the deformation modeling is summarized with a state machine, and it is illustrated in Figure 3.5.

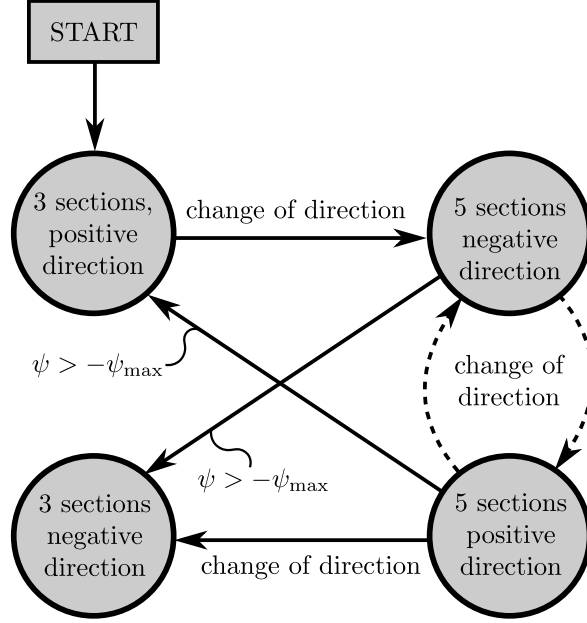


Figure 3.5: State machine representation of the approximation process for the deformation function. The dotted line represents the neglected seven zone fragmentation, which is not a true analytical description, but an assumption.

The state machine in Figure 3.5 is suitable for illustrating the deformation function's analytical approximation, but the scheme was also applied in the simulation environment - presented in section 2.2 - and an animation function was programmed to be able to examine the numerically simulated and the analytically calculated deformations through the simulation time interval.

3.2 Estimation method for the coefficient of friction

After describing the deformation function $q(x, t)$ in the previous subsection, now all the terms are known in the equation of motion - equation (2.14). The expressions for the deformation contain known or observable physical quantities, only the coefficient of friction remain unknown. The definite integral in the equation of motion can be evaluated analytically, because the deformation function is known, if the yaw angle is measured and the history of measurements is stored (the "memory effect" of an elastic tire is considered), an algebraic equation can be formed for the coefficient of friction, as

$$f(\mu) := -J_A \ddot{\psi}(t) - b_t \dot{\psi}(t) + M(t) - \int_{-a}^a k q(x, t) (x - e) dx = 0, \quad (3.7)$$

where $f(\mu)$ is defined as the algebraic expression (which equals zero), that needs to be solved in order to acquire the value of the coefficient of friction. This function cannot be solved analytically, but it is possible to solve with numerical schemes. The chosen numerical solver was the built-in `fsolve` function of Matlab [9], using a trust region algorithm.

As previously discussed, with the help of the created simulation environment, it is possible to test the developed estimation algorithm. The mechanical system was simulated with the physical properties summarized in Table 3.1, then the coefficient of friction can be calculated for each time-step.

Parameter	Notation	Value
Towing length	l	0.01 m
Contact patch half-length	a	0.04 m
Lateral tire stiffness	k	60000 N/m ²
Mass moment of inertia with respect to joint A	J_A	0.1 kgm ²
Torsional damping	b_t	0.4827 Nms
Vertical tire load	F_N	180 N
Coefficient of friction	μ	0.5

Table 3.1: List of parameters used in the numerical simulations for testing the developed estimation algorithm.

In order to test whether the equation for the coefficient of friction (Eqn. (3.7)) is correct, first the value of μ was provided to the algorithm, but the yaw angle was not, therefore the output of the numerical solver was the estimation for ψ . The results can be seen in Figure 3.6. The test was successful, therefore equation (3.7) can be considered accurate. It is now visible that the estimation algorithm can accurately identify the yaw angle. However, ψ can be measured easily, on the other hand the calculation method used the value of the coefficient of friction, which can not be observed.

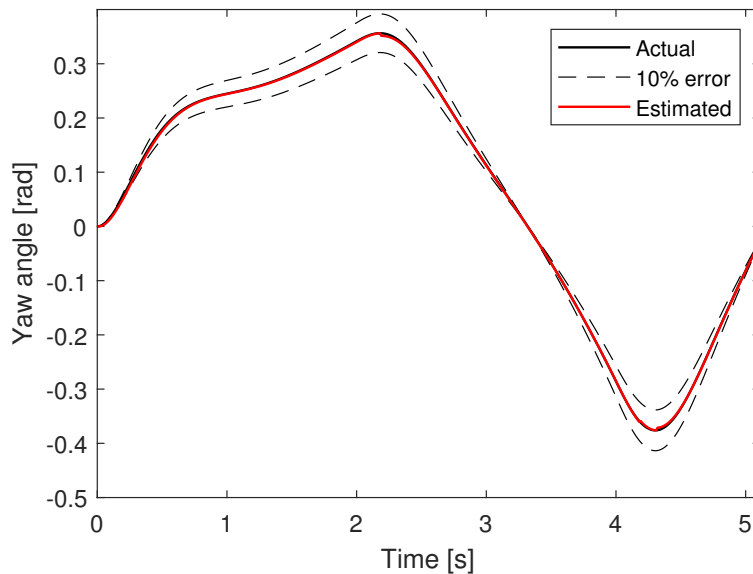


Figure 3.6: Estimating the yaw angle, with the value of the coefficient of friction given to the algorithm. Except small oscillations around the peaks of the true yaw angle function, no visible differences can be seen between the actual and the estimated signals.

The next step is to check, whether the developed estimation algorithm can identify the coefficient of friction correctly. In equation (3.7), after expressing the definite integral, the coefficient of friction appears in nonlinear terms, thus it is harder to solve the equation numerically for μ . With the same physical properties, the result of the estimation can be seen in Figure 3.7. The estimation process was also tested for simulation cases with $\mu = 1$ and $\mu = 1.8$.

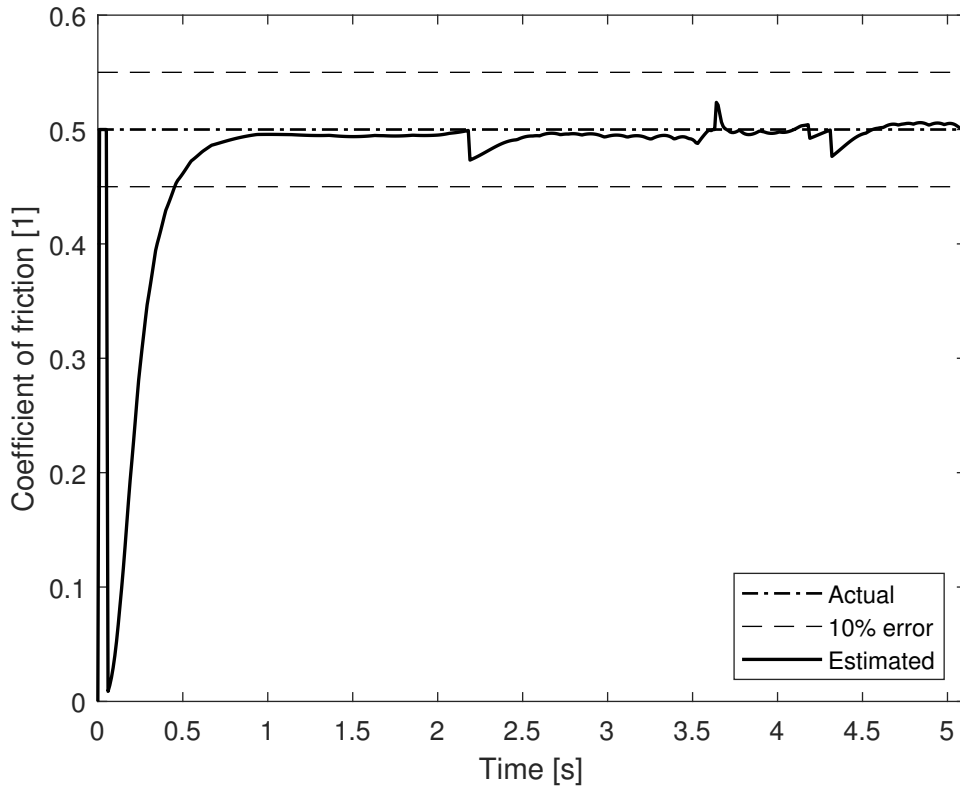


Figure 3.7: Estimation results for the coefficient of friction using simulated data sets.

In Figure 3.7 big errors are visible in the beginning (approx. until $t = 0.5$ s), but those errors can be explained with the small angle differences. In the beginning, only the two sliding ends of the contact line are dependent on the friction. When small angle differences occur, the terms that depend on the friction are so small, that it is hard for the numerical solver to accurately approximate the coefficient of friction. Apart from that, only small oscillations are visible, which occur when changing the direction of the angular motion. If five (or more) segmentation can be found in the contact region, the memory effect of the tires needs to be considered, therefore $f(\mu)$ depends on the local minima/maxima of the yaw angle (previously defined as $\psi_{\max}$). With the time discretization, the true local minimum or maximum cannot be determined, but only approximated, that is the reason of those errors.

Another problem occurred, which originates from the numerical precision of the calculation. In Figure 3.8, $f(\mu)$ is illustrated with the sought value of $f(\mu) = 0$. On the left, a "normal" case is visible, where the numerical solver had no problem with finding the

appropriate solution. On the right side of the figure, a "problematic" case is visible. The function never reaches the value of zero, because of the numerical inaccuracies, therefore the solver approximates the solution with the point of the curve, which is nearest to the x -axis.

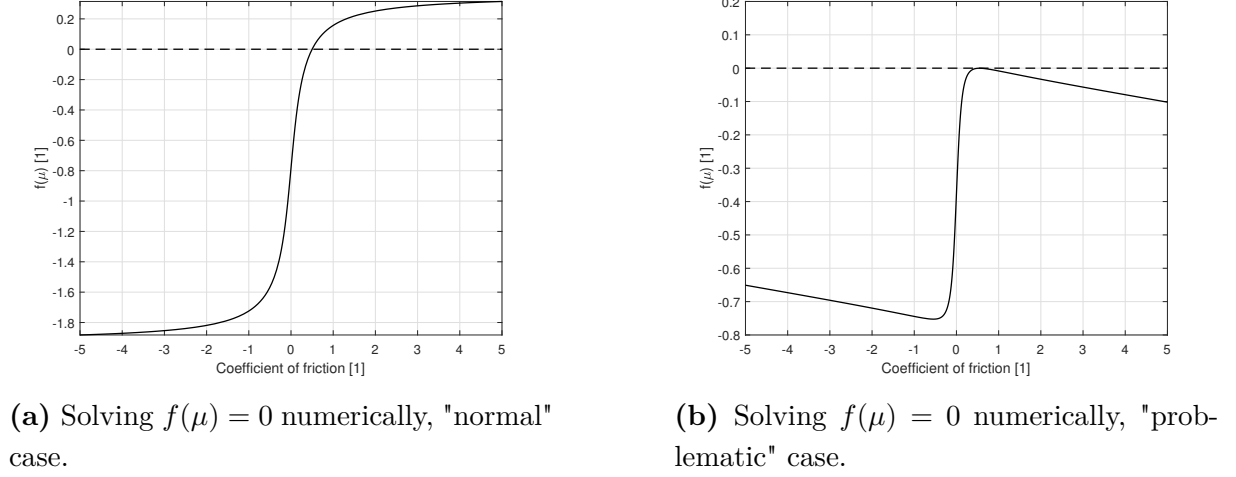


Figure 3.8: Illustrating $f(\mu)$ at two different time instances. The dashed line represents the sought $f(\mu) = 0$ value.

After identifying all the causes of inaccuracies, the same graph as in Figure 3.7 are shown with the noted errors in Figure 3.9.

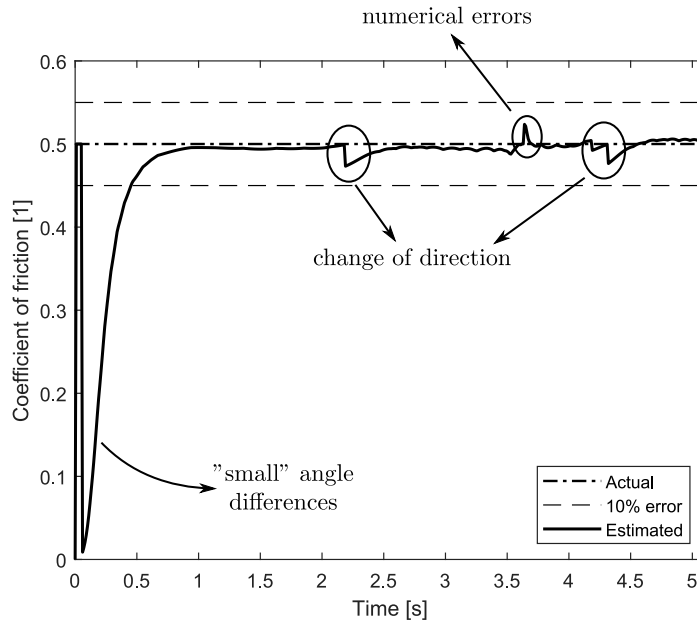


Figure 3.9: The results of the estimation showing the causes of inaccuracies.

Although the proposed estimation process has its limitations and some inaccuracies, it is able to estimate the coefficient of friction for simulated data sets. The biggest advantage of the method is that the results and the errors are easy to interpret, because the method is based on a physical model, the equations are derived from physics-based expressions.

4 Estimation of the coefficient of friction using machine learning

4.1 Estimating with neural networks

4.1.1 A brief introduction to artificial neural networks

Nowadays, machine learning algorithms and artificial neural networks (ANNs, or just shortly NNs) are well-known terms for the majority of academic society. The idea of creating computational systems based on neurons in the human brain, was first introduced about 60 years ago, but in the last 35-40 years, the usage of previously mentioned algorithms spread world-wide [10]. The main reason for the growing applications is that the growing computational power made it possible.

The most important property of ANNs, is the capability to learn complicated connections between patterns from (measured) data. The main application fields are classification or regression tasks, moreover, predictions of time-series events are even possible. The usage of neural networks can be divided into two phases. First, during the learning phase, the final operation of the system is formed, by processing given data structures. During the second phase, the evolved working of the network can be utilized for the prescribed purpose [11].

The basic building blocks of neural networks are so-called neurons, which form layers within the network. The layers can be classified into three main classes, based on their positions in the network. The first layer is called input layer, the last is the output layer, and all the other ones are called hidden layers. A typical representation of this structure can be seen in Figure 4.1. The duty of the input layer is to forward the incoming data, this layer does not take part in the processing of information. The output layer generates an output for each inherent neuron, therefore the number of neurons must match the output of the given task. The number of neurons of each layer and the connection between them can be modified almost arbitrarily.

The layers can also be classified based on their function. Layers are usually divided into categories based on their connection with other layers and how the output of the given layer is calculated. A few common examples are listed below.

- Dense layer: the output is calculated as the linear combination of the input and a constant weight vector.
- Recurrent layer: the layer receives some of the previously calculated outputs with the current input. The memory effect can be modeled this way.
- Convolutional layer: in contrast with the dense layer, the neurons inside a convolutional layer receive only a part of the input.

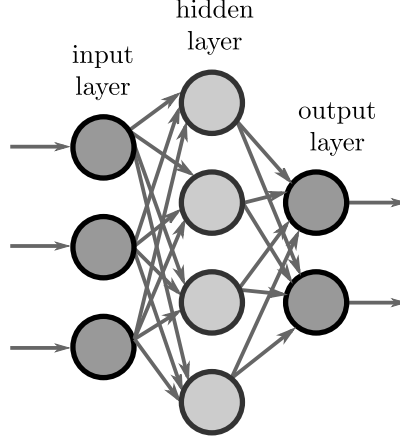


Figure 4.1: Representation of the layers in neural networks.

The connection between distinct layers are calculated with the help of weight vectors and activation functions. Let us consider a NN with L layers, then the output of an arbitrary k^{th} layer ($1 < k < L$) can be expressed as

$$\mathbf{y}_k = \mathbf{h}_{\text{act},k} (\mathbf{W}_k \cdot \mathbf{y}_{k-1} + \mathbf{b}_k), \quad (4.1)$$

where $\mathbf{h}_{\text{act},k}$ is the activation function of the given layer, $\mathbf{W}_k$ is the weight matrix of the layer, $\mathbf{y}_{k-1}$ is the output of the $(k-1)^{\text{th}}$ layer and $\mathbf{b}_k$ is the bias vector of the layer. Assuming that the k^{th} layer has N_k neurons, similarly the $(k-1)^{\text{th}}$ layer consists of N_{k-1} neurons, the output vectors $\mathbf{y}_k \in \mathbb{R}^{N_k}$ and $\mathbf{y}_{k-1} \in \mathbb{R}^{N_{k-1}}$. Also the weight matrix and the bias vector $\mathbf{W}_k \in \mathbb{R}^{N_k \times N_{k-1}}$, $\mathbf{b}_k \in \mathbb{R}^{N_k}$. The relation between the inputs and the output of a single neuron is illustrated in Figure 4.3. Thus, the output of the entire neural network can be expressed as a nested function

$$\mathbf{y} = \mathbf{y}_L = \mathbf{h}_{\text{act},L} \circ \cdots \circ \mathbf{h}_{\text{act},2} \circ \mathbf{h}_{\text{act},1}(\mathbf{x}), \quad (4.2)$$

where $\mathbf{x} \in \mathbb{R}^{N_0}$ is the input vector of the network [4]. In the learning phase, the aim is to find the optimal elements of the weight matrices and the bias vectors in order to minimise the difference between the calculated output of the system and the expected, pre-defined output. For activation functions, usually nonlinear functions are chosen. Typical examples are rectified linear (relu, noted with r), sigmoid (in other words logistic, noted with σ) or hyperbolic tangent (tanh) functions. Mathematically, these functions can be expressed as,

$$r(x) = \max(0, x), \quad (4.3)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (4.4)$$

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1. \quad (4.5)$$

The above mentioned activation functions are illustrated in Figure 4.2.

In the field of machine learning, several learning algorithms are used, usually optimisation techniques, that minimise the difference between the user-defined output of the

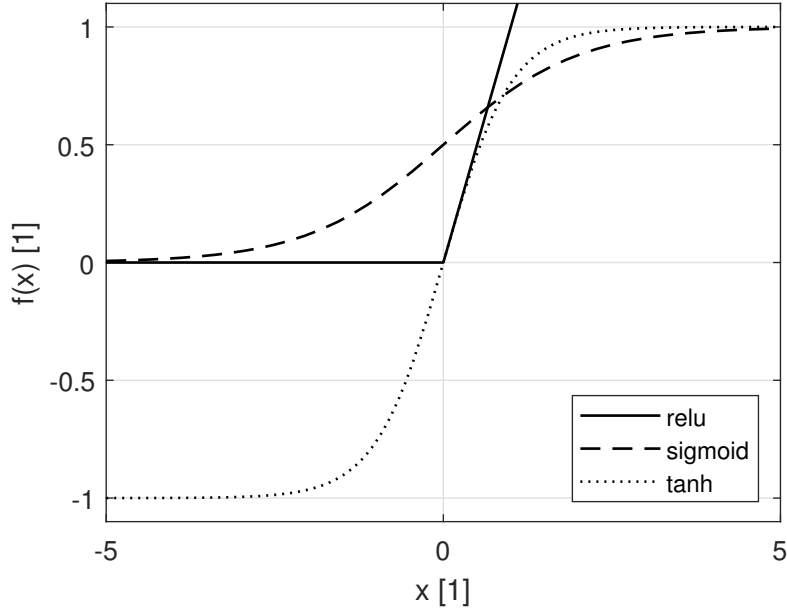


Figure 4.2: Different activation functions illustrated.

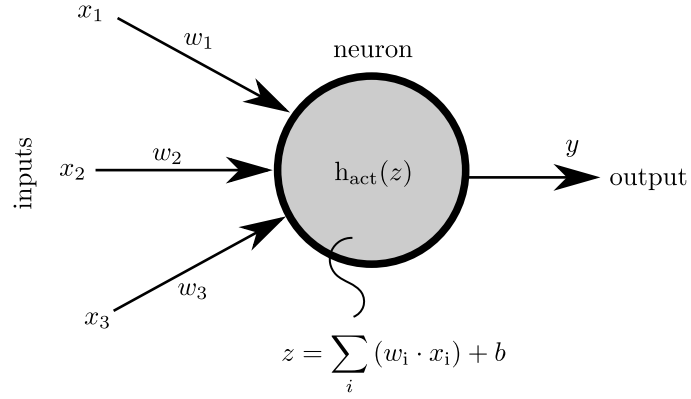


Figure 4.3: A single neuron with three inputs. The inputs, input weights, activation function, bias and output all illustrated.

system and the actually calculated output. Neural networks are often used with iterative optimisation algorithms, for example the gradient descent method. The principle of the method, is a pre-defined cost function, that compares the actual and the predicted output of the system, for example the root mean square error of the data-sets. The parameters of the cost function are the so-called hyper-parameters of the neural network. First, the network initializes the hyper-parameters randomly, then it tunes the parameters iteratively to get closer and closer to the global minima of the cost function. The values of the parameters in the next iteration step are calculated as the difference between the previous parameter vector and the gradient vector multiplied by the learning rate. The learning rate is a constant number, which is defined at the start of the learning process.

An iteration step of the optimisation algorithm is called an epoch, therefore if the gradient descent method found the minima of the cost function with 16 iteration, the

neural network was trained for 16 epochs. But it is not time effective to calculate the gradient for the whole training set, therefore the gradient is only calculated for a small, randomly selected set, called the mini-batch or simply batch, at every iteration step [12].

4.1.2 Estimating the coefficient of friction using neural networks

The neural network was programmed in Python, using the built-in functions from the libraries TensorFlow [13] and Keras [14]. The aim was to create a simple neural network, that can estimate the coefficient of friction without knowing anything from the physical system. The activation functions and the neuron count of the given layers were tuned empirically by checking the results of the estimation for three different activation functions (seen in Figure 4.2). Therefore the structure of the network is quite plain:

- input dense layer with 1024 neurons, sigmoid activation function,
- one dense hidden layer with 512 neurons and also sigmoid activation function,
- output dense layer of 1 neuron with no activation function (the output is the linear combination of the weights and inputs added by the bias of the layer).

The inputs of the neural network at a given time-step was the yaw angle ψ , the previous local minima or maxima of the yaw angle signal $\psi_{\max}$ and the actual steering torque M . The output of the system is the estimated coefficient of friction. The training data set is illustrated in Figure 4.4.

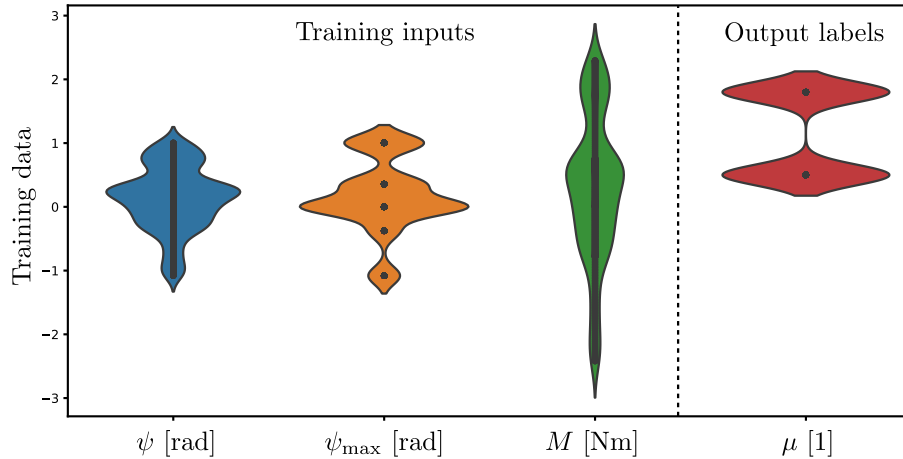


Figure 4.4: Training inputs of the neural network. The violin plot shows the distribution of the data, while the black dots represent the actual data points, therefore it is visible that the yaw angle and the steering torque are continuous functions, while $\psi_{\max}$ and μ are discrete values.

Before setting up the environment for the machine learning part, three different simulations were made using the simulation process introduced in section 2.2. The simulations were made with three different coefficient of friction, with $\mu = 0.5$, $\mu = 1$ and $\mu = 1.8$. The other physical properties remained the same as in Table 3.1. The data sets with the smallest and largest coefficient of friction were used for training the network, the simulation case with the middle value was used for testing the trained neural network.

The training data set was imported in a shared data-frame, and the rows were shuffled in order to prevent over-fitting. The batch-size was selected to be 32, because several studies imply that small batch sizes result in better results with less training time [15]. The network was trained for 30 epochs, using the Adam optimiser from the Keras library with a learning rate of 0.001. During the training, 10% of the data was used for validation to determine the error of the network after each iteration. The error was monitored with the root-mean-squared error (RMSE) metric. A so-called call-back was also defined to restore the weights after the iteration process ended, to match the best validation error achieved during training. The errors of a typical training process can be seen in Figure 4.5. The training error corresponds to the error of the neural network on the data set, that was used for training, while the validation error was calculated for the validation split, which did not take part in training the model.

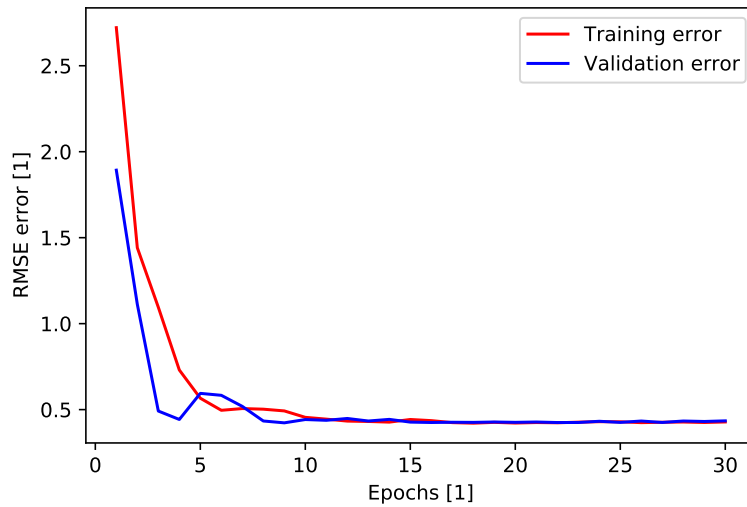


Figure 4.5: Error of the models during a typical training process.

After training the neural network, the accuracy of the estimation was tested on the simulated data-set with $\mu = 1$. The error of the estimation was 0.1068 using the RMSE metric, which is around 10% relative error. The results of the estimation can be seen in Figure below.

A comparison can be made with the derived analytical estimation method in the left side of Figure 4.7. It is visible that the analytical approach (apart from the errors in the beginning) is more accurate than the neural network. But it is important to mention that

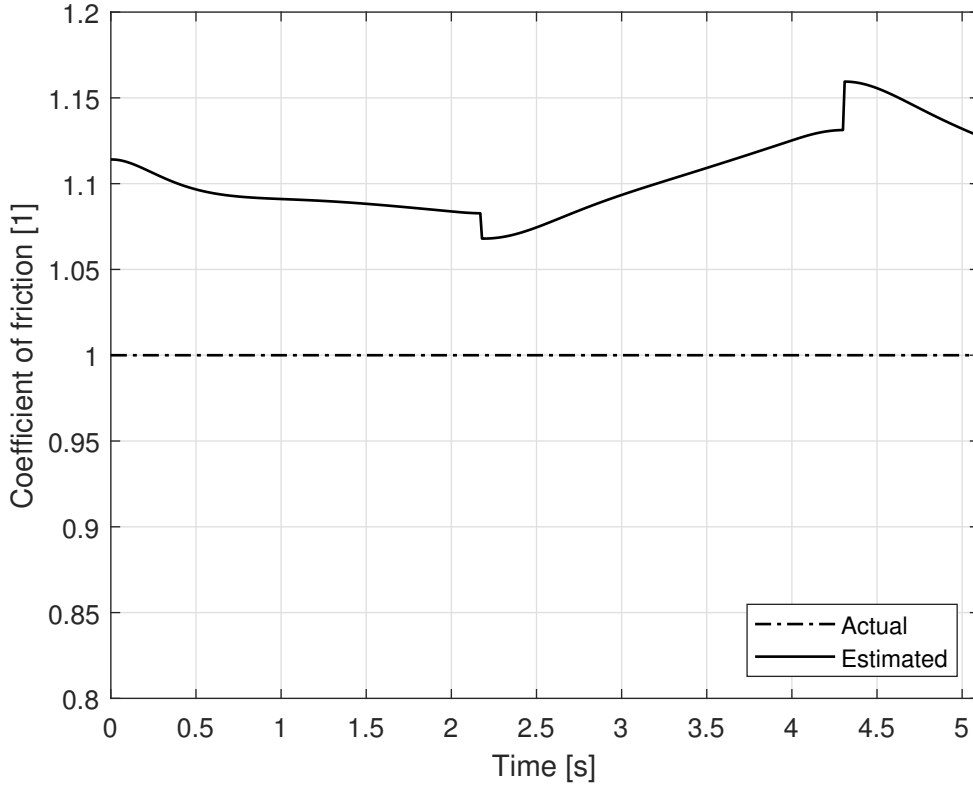


Figure 4.6: Results of the estimation with the simple neural network model.

the test-case was simulated with a steering torque characteristic similar to the simulation cases that were used for training the model. On the left hand side of Figure 4.7, the estimation methods were compared using a different steering torque input. In the end of the simulated time interval, huge errors are visible for the analytical method, because there are more than five fragmentation zones in the deformation function, for which the method was not derived. Also, the neural network estimation is not sensitive to small yaw angles, therefore it provides a more accurate solution in the beginning of the simulation time. Consequently, the proposed analytical estimation method is more accurate, but the neural network is more robust.

4.2 Estimation process implementing a Scientific Machine Learning (SciML) model

While traditional machine learning (ML) algorithms are shown to be able to model non-linear dynamical systems, there are some essential problems that are still not solved. Most importantly machine learning structures need large sets of training data for accurate operation. Although at some cases it is possible to acquire such big data sets via measurements, usually it is not the case. The next problem is that ML models cannot extrapolate out of the training data set well, and lastly the results are hard to interpret. For example, after training a neural network, the weight and bias values can be obtained,

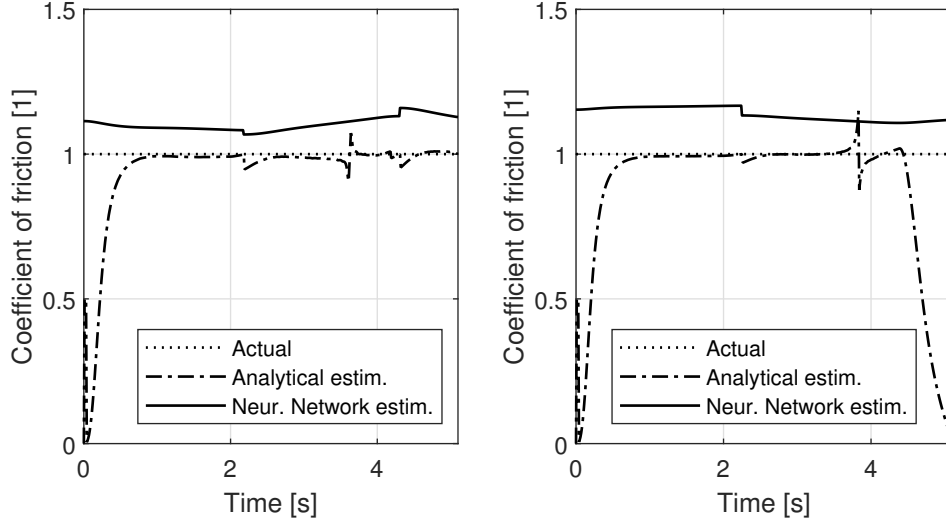


Figure 4.7: Comparing the results of the estimation with neural network with the results from analytical method. On the left the input steering torque characteristic is similar to the steering torque in the training data. On the right different steering torque input was applied, therefore more fragmentation zones have formed in the deformation function, thus the analytical approach was not accurate in the end.

but there is no clear expression that can describe the investigated physical system.

In contrast to traditional ML models, scientific machine learning (SciML [16]) is a quite unique and new field of machine learning researches. The basic principle of the method is to incorporate the knowledge about the investigated physical system, but the effects of the unmodeled terms are compensated with a machine learning structure (e.g. neural networks).

4.2.1 Universal differential equations within scientific machine learning

Within SciML, the use of universal differential equations (UDEs) is a newly applied approach [17]. UDEs are standard differential equations supplemented with machine learnable parts. A non-general form of universal differential equations can be expressed as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{p}) + \mathbf{U}(\mathbf{x}, \mathbf{p}, \mathbf{q}), \quad (4.6)$$

where $\mathbf{x}$ is the unknown vector, $\mathbf{f}$ describes the system with the physics based modeling method, $\mathbf{p}$ vector contains the physical parameters of the mechanical system (assumed to be constant) and $\mathbf{U}$ is the machine learnable part with $\mathbf{q}$ parameters of the machine learning structure (e.g. the hyperparameters of a neural network). Also in equation (4.6) the noise terms are neglected and the state variable $\mathbf{x}$ is assumed to be observable.

A typical work-flow of applying a SciML method with UDEs is to first derive the analytical differential equations that describe the system, then gather measurement data. After fitting the physical properties (noted with $\mathbf{p}$ in equation (4.6)) in a way to match

the physics based approximation with the measurement data the best way. Then finally, the only unknowns are the machine learning model-parameters, that can be obtained by training the model, for example a neural network with gradient descent.

For the model presented in section 2.2, the following UDE system can be derived, but first define the unknown vector as

$$\mathbf{x} = \begin{bmatrix} q(x, t) \\ \dot{\psi}(t) \end{bmatrix}. \quad (4.7)$$

Then from equation (2.12), (2.13) and (2.3) the universal differential equation can be derived, neglecting the nonlinear term:

$$\begin{bmatrix} \dot{q}(x, t) \\ \ddot{\psi}(t) \end{bmatrix} = \begin{bmatrix} (e - x) \cdot \dot{\psi}(t) \\ - (b_t/J_A) \cdot \dot{\psi}(t) - (1/J_A) \cdot \int_{-a}^a k q(x, t) (x - e) dx + (1/J_A) \cdot M(t) \end{bmatrix} + \mathbf{U}(\mathbf{x}, \mathbf{p}, \mathbf{q}), \quad (4.8)$$

where $\mathbf{p}$ vector contains e , b_t , J_A , a and k .

Using the UDE structure in (4.8), with appropriate measurement data, the machine learning algorithm parameters could be calculated, but there is a problem originated from the assumption, that the state variable of the system is observable, which is not the case. Experiments are showed, that it is possible to examine the contact line parameters during static load [18], but this method could not be applied with angular motion, therefore a modification for the UDE SciML method had to be made.

4.2.2 Using scientific machine learning to estimate the coefficient of friction

As previously mention, the SciML method works well with universal differential equations, but due to the nature of the investigated mechanical system, the UDE approach cannot be applied. Thus, a new method was developed, similar to the UDEs, but for nonlinear algebraic equations, that express a physical meaning. Using the derived analytical approach in section 3, a universal nonlinear algebraic equation can be defined as

$$\mu = n(\mathbf{p}) + U(\mathbf{p}, \mathbf{q}), \quad (4.9)$$

where n is the numerical solution of $f(\mu) = 0$, $\mathbf{p}$ vector contains all the physical parameters that need to be provided in order to solve the nonlinear equation $f(\mu) = 0$ and $\mathbf{q}$ is the parameter vector of the machine learning system.

Solving the physics-based part happens the same way, as discussed in section 3. The neural network based part of the solution was determined with a similar machine learning model as in section 4.1.2, the only difference is that the network gets all of the physical parameters of the system as an input, therefore a more robust architecture was found to be more efficient. The activation function was also determined empirically, but in this case the tanh function seemed like a better choice. The structure can be summarized as

- input dense layer with 2048 neurons and tanh activation function,
- one hidden layer with 1024 neurons and also tanh activation function,
- output dense layer (1 neuron), with no activation function.

The model is trained with simulation data sets of $\mu = 0.5$ and $\mu = 1.8$, while the test case is the $\mu = 1$ simulation. Similarly to the neural network validation, another test-case was defined, for a steering torque characteristic, where more than five fragmentation appear in the deformation function in order to test, how will the model react to a case, where the analytic approach is not derived. The results are shown in Figure 4.8.

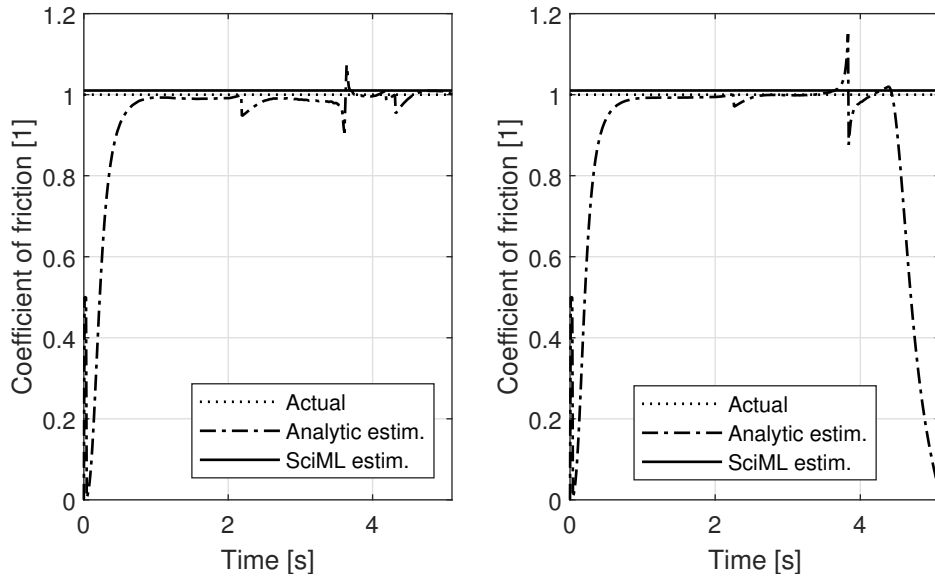


Figure 4.8: The results of the coefficient of friction estimation with the proposed SciML method. The method is tested on a steering torque characteristic, for which the analytic estimation was derived (left), and an other steering torque input for which the analytic method is not accurate (right). The results the analytic estimation method are also shown.

Consequently, as Figure 4.8 also shows, the proposed SciML method's accuracy is compatible with the analytic estimation's, moreover the numerical errors and the unmodeled effects can be also compensated, thus this technique can be utilized to provide a robust and accurate estimation of the coefficient of friction.

5 Measurement

To validate the previously proposed estimation methods, it is not enough to test with simulated data, therefore a measurement set-up needs to be developed. Fortunately experiments have been carried out for a long time at the Department of Applied Mechanics to investigate the dynamical properties of towed wheels [3]. Thus, an already established experimental set-up was used with some modifications, that will be detailed in this section.

5.1 Measurement set-up

At the Department of Applied Mechanics an experimental rig has been developed, that was used during this work. The experimental set-up can be seen in Figure 5.1. The investigated bicycle wheel is mounted in a way that allows the modification of the towing length e . The caster can be rotated about the rigid king-pin, where oil lubrication ensures that the effect of dry friction can be neglected, and low torsional damping is expected. The length of the contact patch can be modified by changing the vertical distance between the king-pin and the "road" surface.

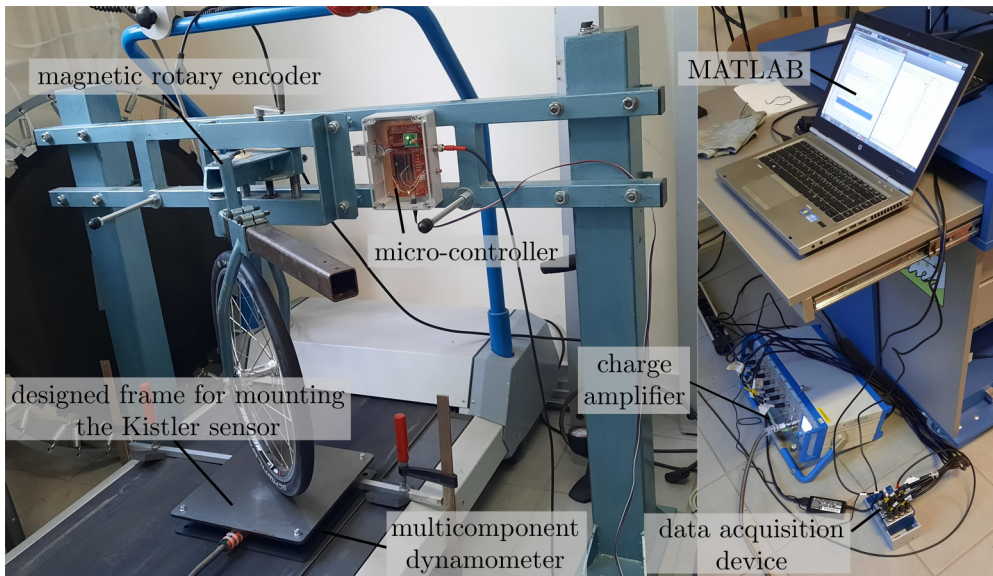


Figure 5.1: Experimental rig with the used devices.

The experimental set-up allows measurements with non-zero towing speed, but the previously derived estimation methods were developed for tires standing still. The steering torque needs to be measured in order to estimate the coefficient of friction, but using this set-up the direct measurement was not possible, because the steering torque was actuated by hand, therefore a Kistler 9129AA Multicomponent Dynamometer was used to measure the self-aligning torque. With zero towing speed, the conveyor belt was not moving, thus it was possible to place the force sensor below the tire. To do so, a special experimental frame was designed and manufactured from metal sheets, to mount the Kistler sensor onto the conveyor belt. For the force sensor a Kistler Type 5080A charge amplifier was

also used. The yaw angle $\psi(t)$ was measured with a magnetic rotation sensor placed at the joint (AMS AS5048A magnetic rotary encoder, 14-bit resolution). For detecting and digitizing the measured angle signals, a PIC micro-controller was used (part of the experimental rig). The measurement data was gathered using a National Instruments USB-6216 (BNC) data acquisition device, but the measured signals were processed in MATLAB.

For testing the estimation methods, different surfaces need to be used during the measurements. The designed frame made it possible to attach an extra plate to which different surfaces could be cemented. Three different plates were selected with different frictional characteristics. One plate was plain steel, the other two plates had emery papers glued on, with different surface roughness.

5.2 Identification measurements

For the analytical estimation model, some physical properties have to be provided. The towing length can be measured on the experimental rig, for all measurements $e = 0.009$ m was preset. The vertical load force was measured by the Kistler 9129AA device, while the length of the contact patch was measured with a vernier caliper. The yaw angle and the self-aligning torque was gathered as previously mentioned.

The only remaining input of the model is the lateral stiffness k , which can be only determined via identification measurements. In practical applications, it would be hard to identify k , but the value of the lateral stiffness is constant for constant internal tire pressure, therefore by measuring the tire pressure, k could be estimated. But in this experiment, the lateral stiffness value is identified from the natural frequency of the system, which can be determined experimentally.

For the neural network based estimation methods, while training the models, an accurate coefficient of friction value needs to be provided. Moreover it is necessary to know the exact value of μ for validation purposes, thus the coefficient of friction also needs to be identified.

5.2.1 Identifying the coefficient of friction for different measurement plates

As mentioned before three different plates were used during the experiments, with three different coefficient of friction. A simple experiment was applied in order to estimate the values of the coefficients of friction. A wooden object with rubber material glued onto the surfaces, was towed on the measured plates, while measuring the forces with the Kistler Multicomponent Dynamometer. The rubber material was chosen to be similar as the elastic rubber of the measured tire. The experiment set-up can be seen in Figure 5.2.

The experiments were carried out with different normal forces for each plate to estimate the coefficient of friction. As mentioned in section 2, the static and dynamic friction



Figure 5.2: Experiment set-up for measuring the coefficients of friction

properties are assumed to be the same, therefore as all three force components were measured (F_x , F_y and F_z), the measured data is fitted to the function expressed as

$$F = \mu N, \quad (5.1)$$

where $F = \sqrt{F_x^2 + F_y^2}$ and $N = F_z$.

The measurement have been carried out for all three plates with different normal loads. The results for one plate (black emery paper) can be seen in Figure 5.3. The fitted curve can be considered accurate, therefore the values of each coefficient of friction (summarized in Table 5.1) are appropriate for validating the estimation methods.

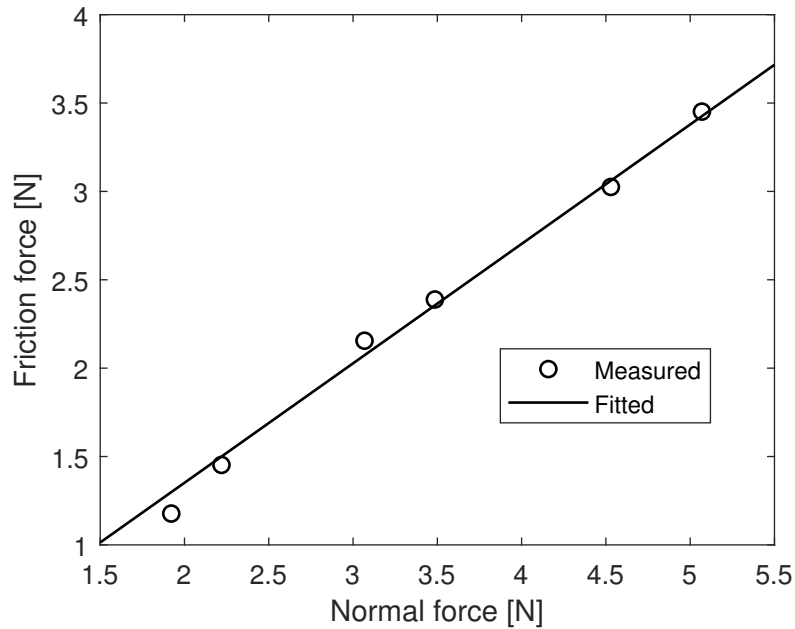


Figure 5.3: Results of the coefficient of friction identification measurement for one of the emery papers.

Plate	Coefficient of friction [1]
Metal plate	1.674
Emery paper (red)	1.9003
Emery paper (black)	0.6757

Table 5.1: Identified coefficients of friction.

5.2.2 Identifying the lateral stiffness of the tire

To identify the distributed lateral stiffness of the tire, the system is modeled as a one degree-of-freedom damped oscillator, then the relative damping ratio ζ and the damped natural angular frequency ω_d was calculated from the measurement data. During the identification measurements the caster was perturbed with an impact, and it was repeated five times for each different contact patch length. The measurement signals for one of the contact patch lengths, can be seen in Figure 5.4.

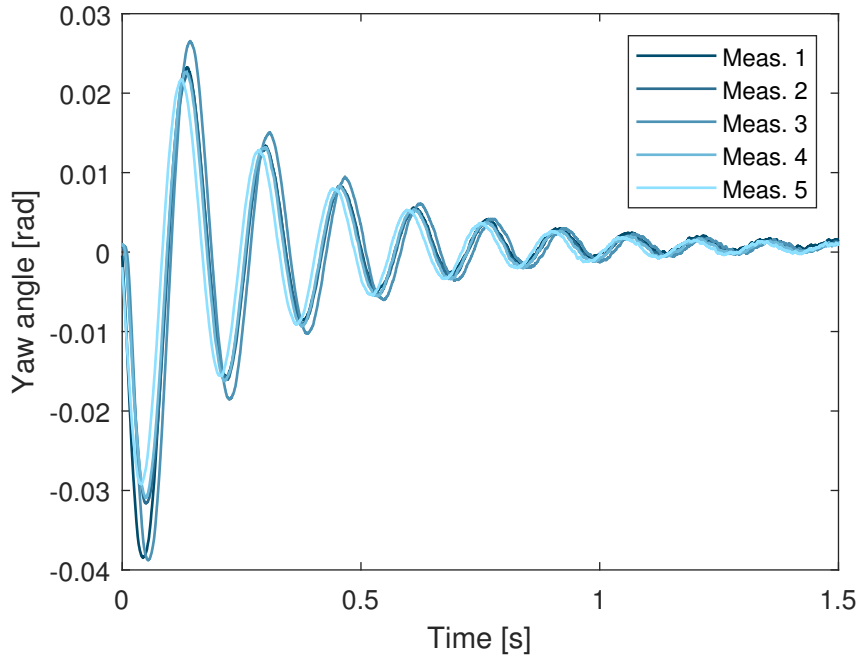


Figure 5.4: Five measurement signals for one of the contact patch lengths. All of the mechanical parameters are the same, only the initial excitation is different, thus the damping and the damped natural frequency are visibly identical.

The natural frequency of the damped system was determined by using the built-in Fast Fourier Transformation (FFT) function in MATLAB, and the natural angular frequency ω_d can be calculated as

$$\omega_d = 2\pi f_d, \quad (5.2)$$

where f_d is the natural frequency of the damped system. The relative damping ratio was

calculated with the method of logarithmic decrements as [19]

$$\Lambda = \frac{1}{n} \ln \left(\frac{A_1}{A_{n+1}} \right), \quad (5.3)$$

$$\zeta = \frac{\Lambda}{\sqrt{4\pi^2 + \Lambda^2}}, \quad (5.4)$$

where Λ is the logarithmic decrement, A_n notes the n^{th} maximal deflection for a given direction and ζ is the relative damping ratio. Then the natural angular frequency of the undamped system ω_n can be determined with the following expression:

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}}. \quad (5.5)$$

Thus, the natural angular frequency of the undamped system and the relative damping ratio is known, now the lateral stiffness value needs to be calculated. From [20], the equation of motion with v towing speed, taking into account the delayed effect of the self-aligning torque can be expressed as

$$J_A \ddot{\psi}(t) + b_t \dot{\psi}(t) + 2ak \left(e^2 + \frac{a^2}{3} \right) \psi(t) = (e - a)vk \int_0^{\frac{2a}{v}} (e - a + v\tau) \psi(t - \tau) d\tau, \quad (5.6)$$

with the previously used notations and τ is the time delay. Thus the natural angular frequency of the undamped system can be expressed as

$$\omega_n = \sqrt{\frac{2ak \left(e^2 + \frac{a^2}{3} \right)}{J_A}}. \quad (5.7)$$

As well as the relative damping ratio:

$$\zeta = \frac{b_t}{2\omega_n}. \quad (5.8)$$

The experiment was repeated with seven different contact patch lengths, while keeping the towing length e the same. The value of k and b_t was calculated to fit the theoretical curves of (5.7) and (5.8) to the measurement results with the least error possible. The results of the experiment and the curve fitting can be seen in Figure 5.5.

While the undamped natural frequency can be identified quite accurately with the FFT algorithm, measuring the damping is much harder, therefore the deviation of the damping data set is larger compared to the natural angular frequency. In Figure 5.5, some outlier values are visible in the measured relative damping. Although by deleting those outlier values, probably a more accurate torsional damping value could be calculated, but the aim of this experiment was the identification of k , therefore the results were accepted and the identified values are summarized in Table 5.2.

All the remaining measurements were carried out with the same inner tire pressure of $p = 1$ bar, therefore the identified distributed lateral stiffness value is valid for all the latter experiments.

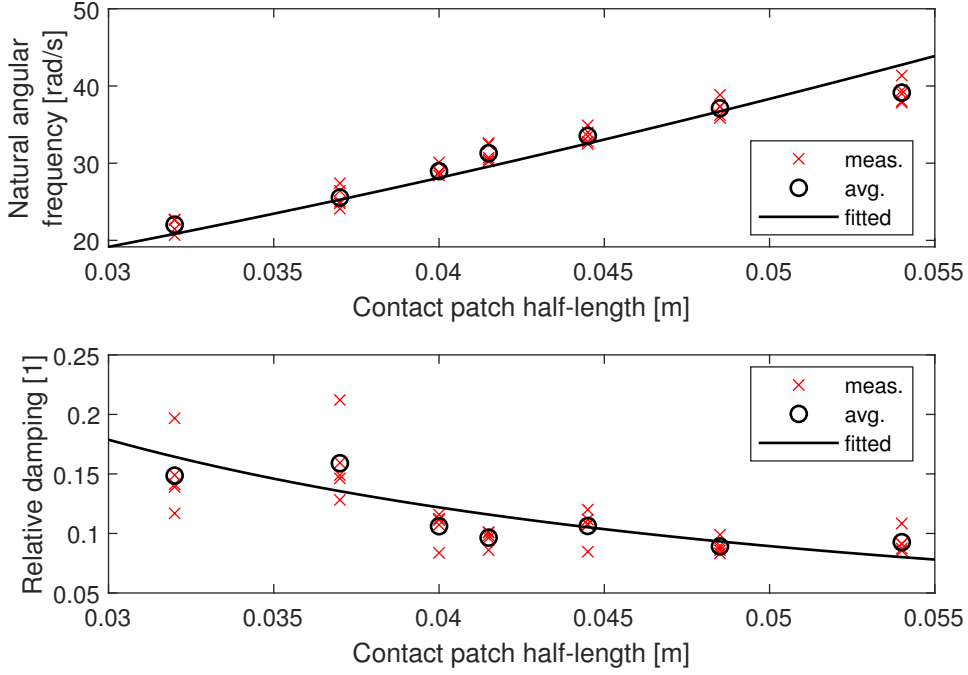


Figure 5.5: Identification results for the lateral stiffness and the torsional damping. The red crosses represent the measured data points, while the black circles are the average value for each contact patch half-length. The black line shows the fitted theoretical curve.

Physical property	Notation	Identified value
Distributed lateral stiffness	k	$2.438 \cdot 10^6 \text{ N/m}^2$
Torsional damping	b_t	6.8509 Nms

Table 5.2: Identified lateral stiffness and torsional damping.

5.3 Evaluating the proposed calculation methods using the measured data-sets

After the identification experiments, the validation measurements were carried out in order to test the proposed estimation methods. While the derived analytic calculation method can be applied for every measurement case, the machine learning based techniques need training data as well, therefore the surfaces with the smallest and the largest coefficient of friction (the black and red emery papers) are used for training the models, then the results are validated on the third measurement case.

The measurements were carried out as discussed in section 5.1, then the post processing was performed in MATLAB. First the angle, force and torque offsets were determined and the signals were corrected with it. The measurement was executed with 4000 Hz sampling frequency, but the evaluation of each data point would take too much time, therefore the signals were under-sampled to match the simulation frequency. Also the angle signal needed to be filtered with a moving average filter, to provide smooth angle signals to the

estimation methods.

The analytic estimation algorithm expects the previous local minima or maxima of the yaw angle (noted with $\psi_{\max}$) as well as the number of fragmentation and the direction of angular motion at every time-step, therefore these values were calculated during the post processing of the data. In Figure 5.6, the yaw angle signal of an arbitrary measurement case is visible with the determined $\psi_{\max}$. In the first zone the value of $\psi_{\max}$ is defined as zero, because there is no local extremum before the first direction change.

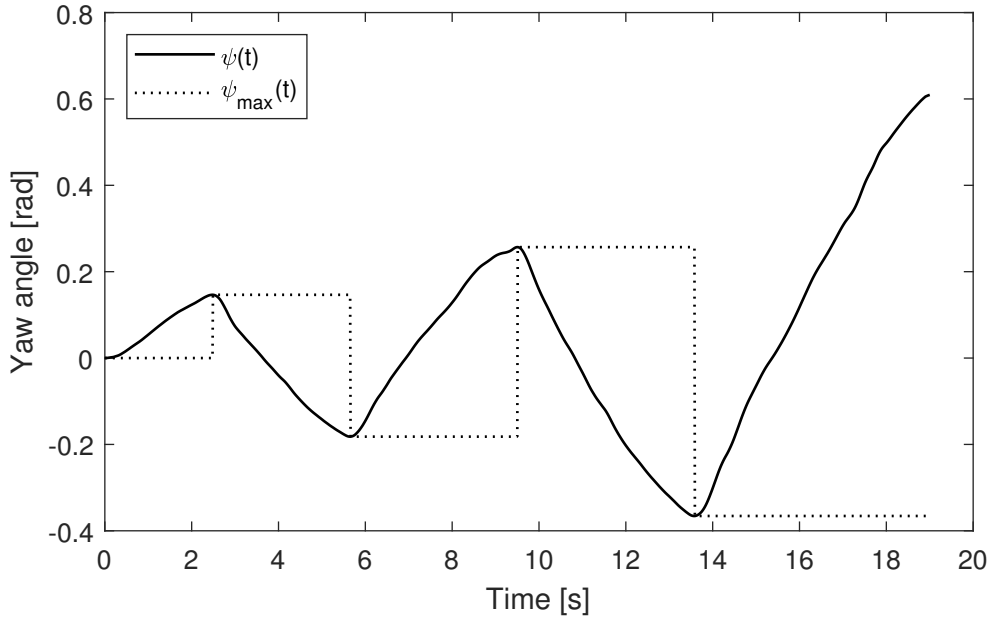


Figure 5.6: The yaw angle signal and the determined $\psi_{\max}$ values of an arbitrary experiment case.

The direction of motion could be easily determined, but for the fragmentation number, the previously proposed state machine (Figure 3.5) was used. The results are visible in Figure 5.7.

5.3.1 Estimation based on the analytic modeling approach

All of the necessary physical or nonphysical input properties have been determined for the analytic coefficient of friction estimation by measurements or during post processing the measured data. Thus, the results of estimation can be compared to the identified coefficient of frictions.

The results of the estimation for both emery papers are reassuring, but the estimation for the metal sheet's coefficient of friction was quite inaccurate. The numerical problems causing big oscillations are similar to the ones in the simulation phase, but the errors caused by small angles and the sudden jumps in case of direction change disappeared. Although the previously identified coefficient of friction values (Table 5.1) seemed accurate,

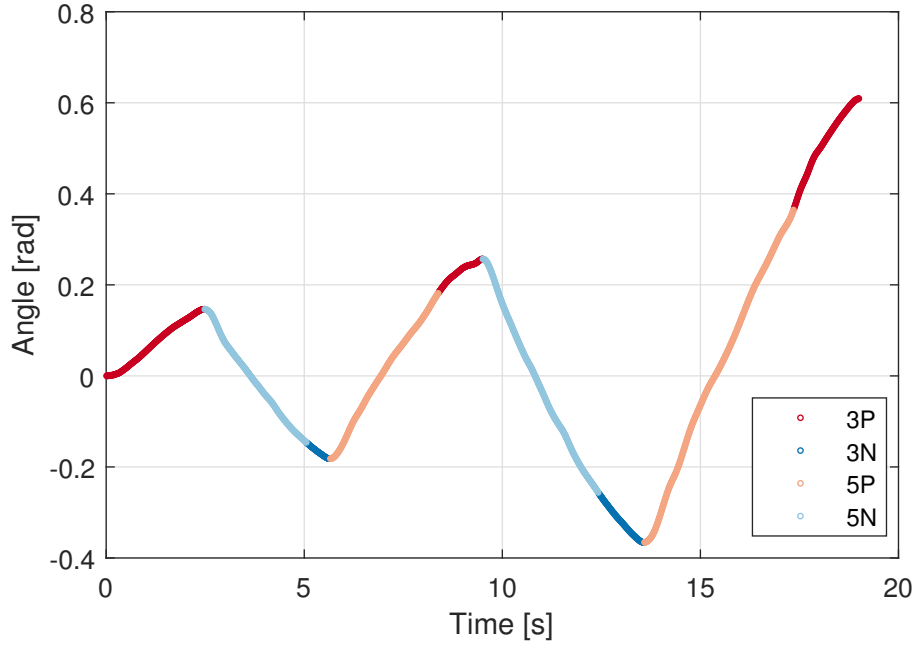


Figure 5.7: The fragmentation number and the direction of angular motion for an arbitrary measurement case. On the legend, the number represents the fragmentation count and the "P" stands for positive, "N" is for negative direction.

there probably is some inaccuracy in the measurement, because the estimated friction coefficient values often tend to the same number, but not to the identified value. An example is shown in Figure 5.8 for one of the estimation cases.

Apart from oscillations caused by the numerical error, the value of the estimation tends to be around $\mu = 0.95 \dots$, but the identified coefficient value is 0.6757. To compare the accuracy of the estimation, the RMSE value was calculated for each estimation. Also the time spent on calculating the estimated values were also monitored with a dimensionless value as

$$\vartheta = \frac{t_{\text{calculation}}}{t_{\text{measurement}}}, \quad (5.9)$$

where $t_{\text{calculation}}$ is the time cost of the estimation and $t_{\text{calculation}}$ is the duration of the measurement. The duration of each calculation was measured on the same computer, with an Intel(R) Core i5-8250U processor, 16 GB of RAM and an NVIDIA GeForce MX150 GPU. The accuracy and the time cost of the estimation is illustrated in box charts in Figure 5.9, where the horizontal blue lines inside each box represent the median of the data-set, the upper and lower ends of each box show the upper and lower quartiles and the horizontal black lines outside of the boxes correspond to the maximum and minimum of the data set.

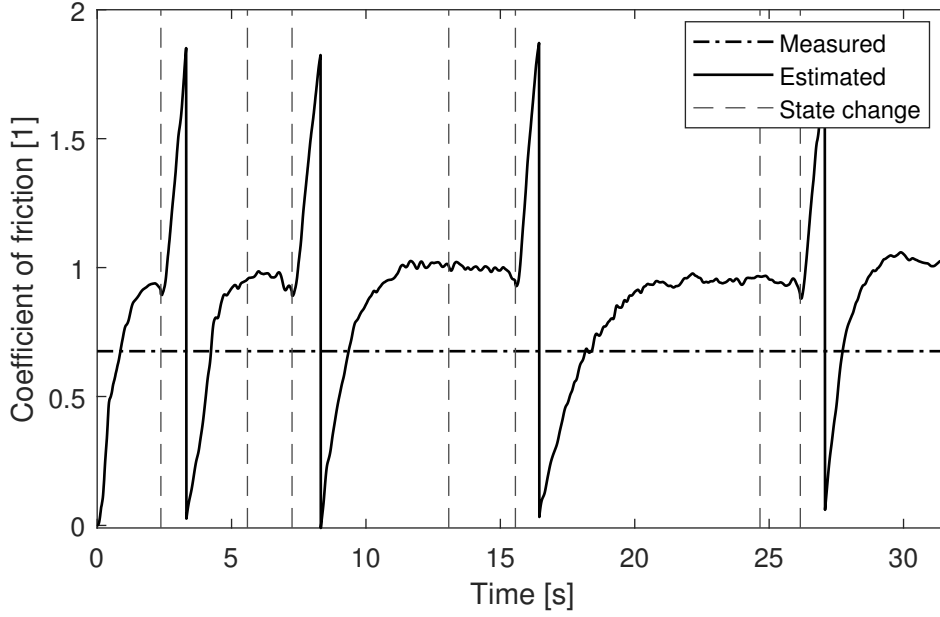


Figure 5.8: Results of the estimation for one of the measurement cases. The vertical dashed lines represent the change of state, according to the state machine derived previously.

5.3.2 Estimation with machine learning based methods

The machine learning based estimations were carried out with the same process. The training data set consisted of the measurements of the black and red emery paper, while the results were tested on the metal sheet measurements. Although the SciML method also considered the solution of the analytic estimation as well as other physical properties, the principle of the method is similar to the neural network estimation. After data post-processing and training the model, the estimation was made for the validation data sets. The training losses can be seen in Figure 5.10 for the training of the neural network model (the training of the SciML model produced similar loss values, but the convergence was faster). In Figure 5.10, it is also visible, that the results converged after around 5 epochs, but the training lasted for 30 epochs. As mentioned before, a call-back has been also implemented to restore the weights to match the best validation accuracy during training, in order to prevent over-fitting.

The results for a chosen measurement case can be seen in Figure 5.11. The analytic estimation is also shown, to be able to compare the results of all three methods.

It is visible from Figure 5.11, that the analytic estimation results could be the most accurate of the three methods, but the huge oscillations caused by the numerical calculation make the estimated signal noisy. Although the SciML estimation is not inside the interval of 10% relative error, the estimation is much smoother and the accuracy can be considered accurate (taking into account that in the literature there have been no better estimation methods found). And lastly, the neural network estimation provides a smooth,

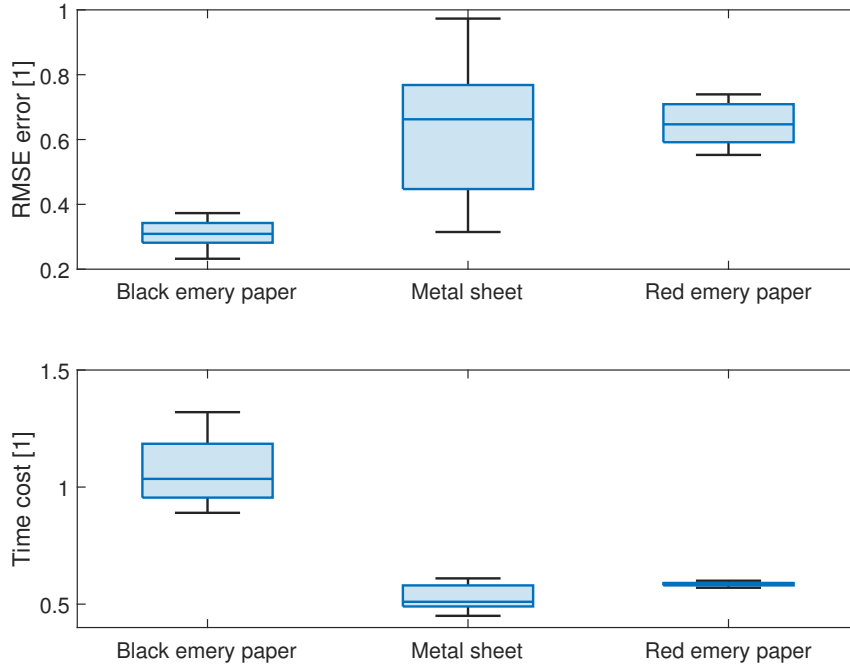


Figure 5.9: Box chart illustration of the estimation error and the time cost with the defined dimensionless time cost quantity.

steady signal, however the SciML implementation improved the results by a significant amount. The averaged RMSE error of the methods for all validation measurements are summarized in Figure 5.12, as well as the computational cost of each method.

Although the error of the analytic estimation can be reduced to nearly half of their value using a SciML method, it is worth mentioning that the simple neural network approach provides moderately accurate estimation with much less computational time, therefore could be ideal for practical applications with real-time data processing.

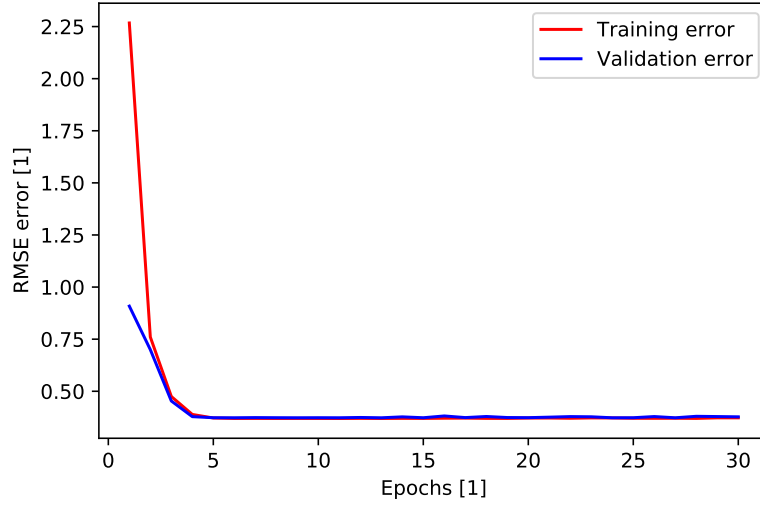


Figure 5.10: Error of the neural network model during training on the measurement data.

6 Summary and conclusion

The goal of this work was to develop a method that is able to estimate the coefficient of friction between the road surface and elastic tire. In this work, the longitudinal dynamics of the vehicle were neglected, the focus of the study was to investigate the situation, when the towing speed is zero, thus only the angular motion is considered. A numerical simulation was successfully created that implements the equations provided by the brush tire model, in order to investigate the relationship between the road and tire surfaces. Then an analytic estimation method was developed for estimating the coefficient of friction based on the brush model. Beside analytic approaches, two machine learning based algorithms were used to estimate the coefficient of friction. A simple neural network model, as well as a SciML method, that combines data-driven modeling with the physics based part. All three methods were tested on simulation data, as well as measurement data, gathered by experiments organized by us. An experiment set up was created using already existent measurement equipment as well as custom designed and manufactured gear. The measurement data was post-processed and transformed into such form, that the estimation methods could be evaluated using the gathered data. Finally, the conclusions are drawn and some further improvements can be conceived.

6.1 Results

All three estimation methods were able to estimate the coefficient of friction reliably, using simulation data set. Although the analytic technique showed big oscillations around the desired value, from time to time, the effect of those numerical errors was negligible. The accuracy of the SciML implementation was as good as the accuracy of the analytic

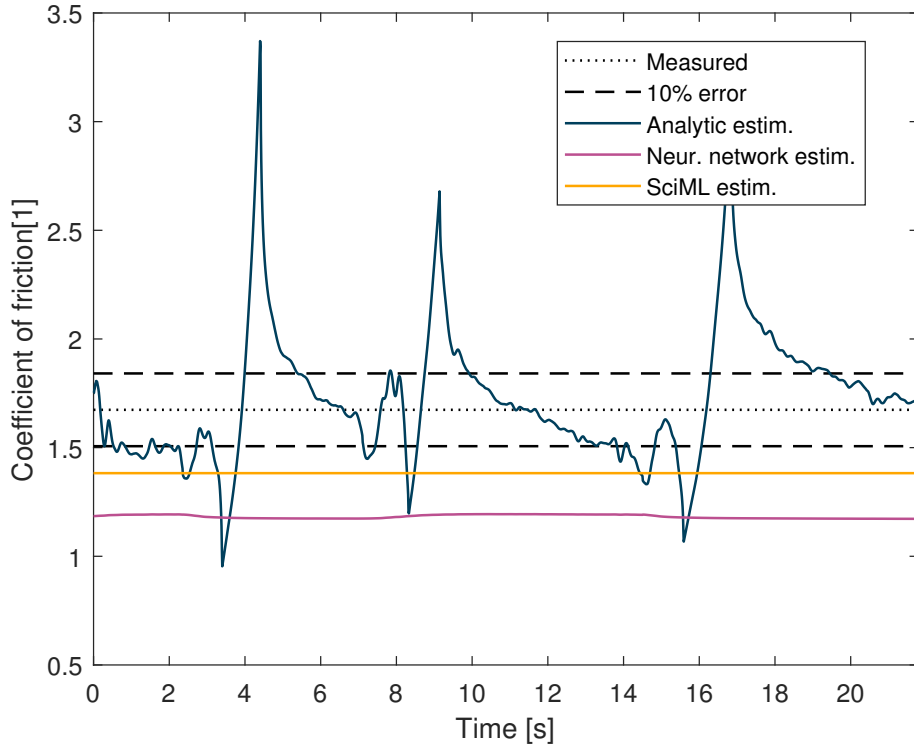


Figure 5.11: Results of the estimation methods for one of the validation measurements. The identified coefficient of friction value with the 10% relative error limits are also shown.

algorithm, but more robust, in terms of operational conditions. The analytic method was only derived for less than five fragmented zones in the deformation function of the contact patch, therefore gave false results when more zones were formed. In spite of this, the accuracy of the SciML method was not declined by that much during these conditions. The classical neural network method was not as accurate as the SciML implementation, but provided reasonable estimation with far less computational time.

On measured data, the numerical errors of the analytic calculation method caused much larger oscillations than previously on simulation data, therefore the accuracy of the method declined. The run time of the calculation was reasonable, but would require more modifications to the program for a possible real-time application. The neural network based estimation proved to be moderately accurate on measurement data, but far less time was needed for the calculations compared to the previously discussed analytic method. And finally, the SciML implementation was the most accurate, but the time cost of the estimation was the largest for this algorithm. The error values, time cost and the robustness properties are summarized in Table 6.1.

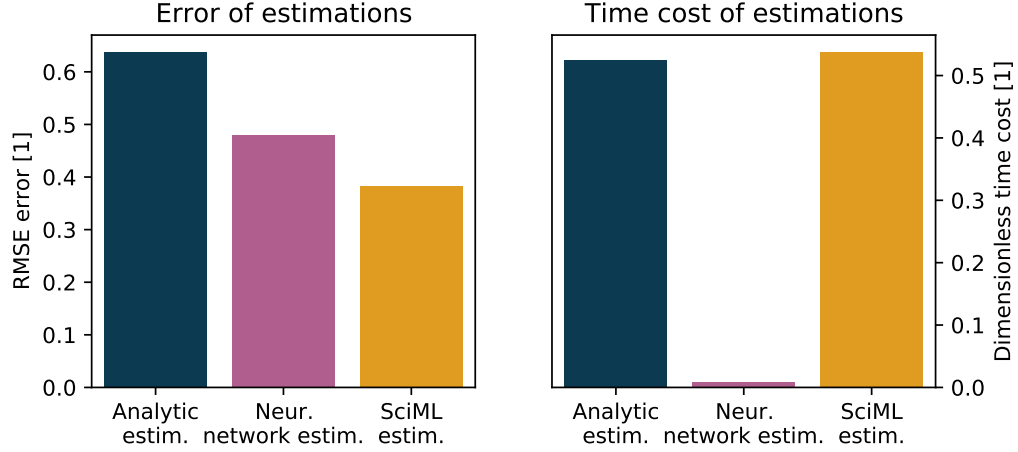


Figure 5.12: Comparing the accuracy of the estimations for the validation measurement cases, as well as the time cost of the computations.

	RMSE error [1]	Time cost (dimensionless) [1]	Robustness
Analytic method	0.638	0.525	for the derived cases
Neural network	0.4794	0.0085	✓
SciML method	0.3837	0.538	~

Table 6.1: Summarized results of each estimation method.

6.2 Outlook, further development

The developed simulation environment modeled the relation of tire and road surface by the brush tire model, which is an accurate model during certain conditions. The performance of the model could be enhanced e.g. with considering different frictional properties in the sticking and sliding region [8]. Another improvement in the simulation accuracy would be to consider a more complex tire model, for example the stretched string model [21], or a modern tire model [22].

To improve the results of the analytic estimation, some modification would need to be made in the model. The linearization of the contact patch deformation function around $\psi = 0$ rad made the comparison of the simulation and the analytic calculation more convenient, but the estimation results could be improved by taking into account the nonlinear form of deflections.

The accuracy of the neural network models was optimized by tuning the model parameters of the networks, but the chosen architecture was not modified after a given complexity. The performance of these machine learning models could be even more improved by implementing complex network structures, but the aim of this study was to demonstrate the potential application of neural networks with relatively plain structures. Although the models worked quite well for simulation data, the nominal accuracy was not reached for validation measurements. The literature suggest the use of other machine learning structures (e.g. Gaussian processes [23]) for measurement data sets, because the

neural network based models may not be able to provide accurate solutions for noisy measurement data [4].

Studies have showed that vehicle path prediction [24], vehicle motion control [25] and antilock braking system (ABS) control [26] are more efficient if the estimated coefficient of friction signal is available. Moreover, the end goal of this work was to provide a method for estimating the coefficient of friction, in order to use the knowledge about the frictional characteristics for robust steering control. For precise steering control, the understanding of self-aligning torque (dependent on the friction) is essential, thus the proposed estimation methods provide a solid foundation for the above mentioned applications.

Reference

- [1] J. Svendenius, *Tire Modeling and Friction Estimation*. PhD thesis, Department of Automatic Control, 2007.
- [2] H. B. Pacejka, “Chapter 3 - theory of steady-state slip force and moment generation,” in *Tyre and Vehicle Dynamics (Second Edition)*, pp. 90–155, Oxford: Butterworth-Heinemann, 2006.
- [3] S. Beregi, D. Takacs, G. Gyebroszki, and G. Stepan, “Theoretical and experimental study on the nonlinear dynamics of wheel-shimmy,” *Nonlinear Dynamics*, vol. 98, p. 2581–2593, Sep 2019.
- [4] S. Beregi, D. Barton, D. Rezgui, and S. Neild, “Using scientific machine learning for experimental bifurcation analysis of dynamic systems,” workingpaper, Oct. 2021.
- [5] F. Holzmann, M. Bellino, R. Siegwart, and H. Bubb, “Predictive estimation of the road-tire friction coefficient,” in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 885–890, 2006.
- [6] T. Umeno, E. Ono, K. Asano, S. Ito, A. Tanaka, Y. Yasui, and M. Sawada, “Estimation of tire-road friction using tire vibration model,” *SAE Transactions*, vol. 111, pp. 1553–1558, 2002.
- [7] J.-O. Hahn, R. Rajamani, and L. Alexander, “Gps-based real-time identification of tire-road friction coefficient,” *Control Systems Technology, IEEE Transactions on*, vol. 10, pp. 331 – 343, 06 2002.
- [8] A. O’Neill, J. Prins, J. F. Watts, and P. Gruber, “Enhancing brush tyre model accuracy through friction measurements,” *Vehicle System Dynamics*, vol. 60, no. 6, pp. 2075–2097, 2022.
- [9] MathWorks, “Solve system of nonlinear equations - MATLAB.” https://www.mathworks.com/help/optim/ug/fsolve.html?s_tid=srchtitle_fsolve_1. Accessed: 2022-10-29.
- [10] M. Altrichter, G. Horváth, B. Pataki, G. Strausz, G. Takács, and V. József, *Neurális hálózatok*. Panem Kft., 2006.
- [11] Á. Antal and B. M. Györök, “Központi idegrendszer működésének modellezése neurális hálóval egy helyben állás közben,” *Budapesti Műszaki és Gazdaságtudományi Egyetem, Tudományos Diákköri Konferencia*, 2021.
- [12] A. Geron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O’Reilly Media, 2017.

- [13] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI’16, (USA), p. 265–283, USENIX Association, 2016.
- [14] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.
- [15] D. Masters and C. Luschi, “Revisiting small batch training for deep neural networks,” 2018.
- [16] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, K. Willcox, and S. Lee, “Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence,” 2 2019.
- [17] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, and A. Ramadhan, “Universal differential equations for scientific machine learning,” *arXiv preprint arXiv:2001.04385*, 2020.
- [18] D. Takács, G. Orosz, and G. Stépán, “Delay effects in shimmy dynamics of wheels with stretched string-like tyres,” *European Journal of Mechanics A-solids*, vol. 28, pp. 516–525, 2009.
- [19] G. Csernák and G. Stépán, *Rezgéstan*. Budapesti Műszaki és Gazdaságtudományi Egyetem Műszaki Mechanikai Tanszék, 2019.
- [20] D. Takács and G. Stépán, “Experiments on quasiperiodic wheel shimmy,” *Journal of Computational and Nonlinear Dynamics - J COMPUT NONLINEAR DYN*, vol. 4, 07 2009.
- [21] H. B. Pacejka, “Chapter 5 - non-steady-state out-of-plane string-based tire models,” in *Tire and Vehicle Dynamics (Third Edition)* (H. B. Pacejka, ed.), pp. 211–286, Oxford: Butterworth-Heinemann, third edition ed., 2012.
- [22] C. Oertel and A. Fandre, “Rmod-k tyre model system,” *ATZ worldwide*, vol. 103, 11 2001.
- [23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005.
- [24] C.-S. L. G. student and H. P. A. Professor, “Road friction coefficient estimation for vehicle path prediction,” *Vehicle System Dynamics*, vol. 25, no. sup1, pp. 413–425, 1996.

- [25] K. B. Singh and S. Taheri, “Estimation of tire–road friction coefficient and its application in chassis control systems,” *Systems Science & Control Engineering*, vol. 3, no. 1, pp. 39–61, 2015.
- [26] S. Rajendran, S. K. Spurgeon, G. Tsampardoukas, and R. Hampson, “Estimation of road frictional force and wheel slip for effective antilock braking system (abs) control,” *International Journal of Robust and Nonlinear Control*, vol. 29, no. 3, pp. 736–765, 2019.