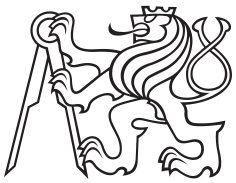
Master's Thesis



Czech Technical University in Prague

FEE

Faculty of Electrical Engineering
Department of control engineering

Vehicle states estimation

Bc. Karim Al Reyahi
Field of study: Cybernetics and robotics
Subfield: Systems and control

Supervisor: Ing. Tomáš Haniš, Ph.D.
May 2019

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Al Reyahi Karim** Personal ID number: **420399**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Systems and Control**

II. Master's thesis details

Master's thesis title in English:

Vehicle states estimation

Master's thesis title in Czech:

Odhadování stavů vozidla

Guidelines:

The goal of the thesis is follow-up development in vehicle IMU platform, namely the development and implementation of state estimation algorithms in Matlab & Simulink environment. The thesis will have two fundamental parts. First task will be reliable measurement and/or estimation of vehicle CG longitudinal and lateral speed and vehicle yaw rate. Where the second part will deal with estimation of tire variables, like tire slip and slip angle, governing forces and moments acting upon vehicle. This is very challenging task dealing with highly non-linear dynamics and time variant system. Results of both parts are essential for further vehicle control system design development.

Bibliography / sources:

[1] Edward M. Kasprzak, L. Daniel Metz, William F. Milliken Douglas L. Milliken, Race Car Vehicle Dynamics - Problems, Answers and Experiments, Premiere Series Books, 2015, ISBN-10: 0768011272.
[2] Prof. Dr.-Ing. Uwe Kiencke Prof. Dr. Lars Nielsen; Automotive Control Systems, Springer, 2005
[3] Jan Filip, Trajectory Tracking for Autonomous Vehicles, 2018
[4] Denis Efremov, Unstable ground vehicles and artificial stability systems, 2018

Name and workplace of master's thesis supervisor:

Ing. Tomáš Haniš, Ph.D., Department of Control Engineering, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **01.02.2019** Deadline for master's thesis submission: **24.05.2019**

Assignment valid until:
by the end of summer semester 2019/2020

Ing. Tomáš Haniš, Ph.D.	prof. Ing. Michael Šebek, DrSc.	prof. Ing. Pavel Ripka, CSc.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

Firstly, I would like to express my huge thanks to my supervisor, Ing. Tomáš Haniš Ph.D., for his valuable advices, comments and ideas during the time I worked on the thesis.
Secondly, I want to thank my colleagues: Ing. Denis Efremov, for his help with the vehicle modelling at the beginning of the project and Ing. Marek László for the data and its interpretation from the FEE eForce formula, along with the formula parameters, which allowed me the comparison of results with real data.
Lastly, but probably the most importantly, I am very grateful for the support of my family and friends during the study period at the CTU in Prague.

Declaration

I hereby declare that I wrote the presented thesis on my own and that I cited all the used information sources in compliance with the Methodical instructions about the ethical principles for writing an academic thesis.

Prague, 20. May 2019

.....

Abstract

The evolution of autonomous vehicles is rapidly progressing forward. In order to achieve fully automotive vehicles, as much data about the vehicle behaviour is needed to be known, as possible. The budget is often restricted and cost is very important for the project. Sometimes, the physical quantities are very difficult to measure. The measurements are very often distorted by a surrounding noise, making the results not clear. These situations are examples of where filtering and estimation algorithms come forward, clearing the noise out of the measurements and allowing us the "prediction" of state values at different levels (full vehicle, tyres). The algorithms act as a virtual sensor, which can save the total budget. The aim of this thesis is to estimate the vehicle states, which are not directly measured at both levels, using a restricted measurement vector and a system model.

At the beginning of the thesis a Single-track vehicle modelling approach is introduced, followed by a data-generation process and an electric formula used for real data validation is presented. Later on, the Kalman filter is described with the extension to Extended Kalman filter for non-linear systems. Then the achieved results are shown with a few methods checking the estimate basic assumptions. In the last chapter a future work is being suggested.

Keywords: Single-track vehicle model, Pacejka Magic Formula, Electric formula, Extended Kalman filter, vehicle states estimation, Kalman filter performance check

Supervisor: Ing. Tomáš Haniš, Ph.D.

Abstrakt

Vývoj autonomních vozidel se velmi rychle posouvá vpřed. Aby bylo možné dosáhnout plně autonomních vozidel, je nutné získat co nejvíce dat, která popisují chování vozidla. Rozpočet projektů bývá omezený, proto hrají celkové náklady důležitou roli. Veličiny by občas mohly být pouze velice obtížně měřitelné a často je měření zašuměné vlivem okolí. Naměřené hodnoty pak nejsou přesné. V těchto případech se využije filtrace a odhadování, které nejdříve odstraní velkou část šumu z měření a zároveň "predikují" hodnoty veličin (jak na úrovni celého vozidla, tak i pneumatik). Algoritmy slouží jako virtuální senzor, který může ušetřit celkové náklady vývoje. Cílem této práce je odhadování stavových veličin vozidla za použití omezeného vektoru naměřených hodnot a modelování systému.

Začátek práce představuje jednostopý model vozidla a vytvoření stavového popisu systému, následně je představena elektrická formule, na které byla naměřena reálná data a modely, které se v práci používají. Dále je vysvětlena funkčnost Kalmanova filtru s rozšířením na tzv. Extended Kalman filter, určený pro nelineární systémy. Poté jsou zobrazeny výsledky simulací a porovnání s naměřenými daty, některé výsledky jsou rozšířeny o testování základních předpokladů pro odhad. V poslední kapitole jsou navržena možná rozšíření této práce.

Klíčová slova: Jednostopý model vozidla, Pacejka Magic Formula, elektrická formule, Rozšířený Kalmanův filtr, odhadování stavů vozidla, vyhodnocení výsledků Kalmanova filtru

Překlad názvu: Odhadování stavů vozidla

Contents

Figures

Acronyms

ADAS	Advanced driver-assistance systems.
CoG	Centre of gravity.
d-q	direct and quadrature axis motor model.
EKF	Extended Kalman Filter.
ESC	Electronic stability control system.
FEE	Faculty of Electrical Engineering.
GPS	Global positioning system.
IMB	Innovation magnitude bound.
IMU	Inertial measurement unit.
KF	Kalman Filter.
LMS	Linear mean-square.
MS	Mean square.
NED	North,East,Down coordinate system.
NIS	Normalised innovation squared.
NRMSE	Normalised root-mean-square error.
PMF	Pacejka's Magic formula.
RPM	Revolutions per minute.
S-T	Single-track.
SNR	Signal-to-noise ratio.
TR	Transmission ratio.

Nomenclature

Symbol	Description	Units
$\alpha_{F,R}$	Front/rear tyre slip angle	rad
β	Side slip angle of the vehicle	rad
$\ddot{\psi}$	Yaw acceleration of the vehicle	$rads^{-2}$
$\ddot{\rho}_{f,r}$	Front/rear wheel angular acceleration	$rads^{-2}$
$\delta_{F,R}$	Steering angle of the front/rear wheel	rad
$\dot{\beta}$	Side-slip rate of the vehicle	$rads^{-1}$
$\dot{\psi}$	Yaw rate of the vehicle	$rads^{-1}$
$\dot{\rho}_{f,r}$	Front/rear wheel angular velocity	$rads^{-1}$
$\dot{v}$	Acceleration of the vehicle	ms^{-2}
$\epsilon(k)$	Measurement error	
$\hat{x}$	Estimate of the state	
λ	Tyre slip ratio	–
μ	Friction coefficient of the road	–
ν	Innovation	
ω	Rotational velocity	$rads^{-1}$
ϕ	Roll angle	rad
ψ	Yaw angle	rad
σ	Standard deviation of a sample	
$\tau_{BF,R}$	Braking torque of the front/rear wheel	Nm
$\tau_{F,R}$	Driving torque of the front/rear wheel	Nm

θ	Pitch angle	rad
$\varepsilon\{.\}$	Mean expected value of an argument	–
$B_{x/y}$	Stiffness factor in Pacejka's Magic formula	–
C_α	Cornering stiffness in Two-line tyre model	–
$C_{x/y}$	Cornering stiffness in Pacejka's Magic formula	–
$D_{x/y}$	Peak factor in Pacejka's Magic formula	–
$e, v(k)$	Measurement and system noise	
$E_{x/y}$	Curvature factor in Pacejka's Magic formula	–
F_x	Longitudinal force acting on the vehicle	N
F_y	Lateral force acting on the vehicle	N
F_z	Wheel load	Nm
$F_{x_{f,r}}$	Longitudinal force acting on the front/rear wheel	N
I_z	Moment of inertia of the vehicle around z-axis	kgm^2
$J_{f,r}$	Moment of inertia of the front/rear wheel	kgm^2
$k_{f,r}$	Road drag coefficient of the front/rear wheel	Ns
$L(k)$	Kalman gain	–
$l_{f,r}$	Distance from the vehicle's CoG to the front/rear wheel	m
m	Mass of the vehicle	kg
M_z	Moment of the vehicle around z-axis	Nm
N	Number of samples	–
P	Power of the motor	W
$P(k)$	State covariance matrix	
q	Normalised innovation	
Q, R	State and measurement noise covariance matrices	
$R_{f,r}$	Radius of the front/rear wheel	m
RPM	Revolutions per minute of the motor	–
TR	Transmission ratio	–
$u(k)$	System input	

NOMENCLATURE

v	Velocity of the vehicle CoG	ms^{-1}
v_{ci}	Circumferential velocity of the wheel	ms^{-1}
$v_{xf,r}$	Front/rear wheel travelling velocity	ms^{-1}
$y(k)$	System output	

Chapter 1

Introduction

Nowadays, the evolution of autonomous vehicles is on the rise. Year after year there are new innovations in the control and safety system development, everything is heading towards the full autonomy. Right now, the vehicle systems like ADAS (Advanced driver-assistance systems) are focused on the best car handling from the safety point of view, helping the driver in unordinary, sometimes critical situations. These systems require lots of data from various quantities. Unfortunately, all the measured data contain some amount of added noise, making it not accurate or sometimes false.

1.1 Motivation and approach

In the case, where budget is concerned, it could be very beneficial to analyse the requirements and needs. There is a possibility of omitting several hardware sensors (and thus saving more money), if the estimation algorithm, acting as a virtual sensor, is accurate and fast enough for a particular project. Not every time it is possible to have a sensor fit into the vehicle for particular physical quantity measurement and sometimes it would be very difficult to measure the desired value. Approaches, that determine the side-slip angle from the side-slip rate integration, are often prone to uncertainty and errors from sensor biases [12]. This is where estimation of vehicle states can be useful for the use in the vehicle active control systems. These are nowadays also parts of the vehicle safety systems or as a performance enhancing systems, as an example can be electronic stability control (ESC), which detects the loss of traction and tries to help in reducing the skidding using the brakes.

As noted in [11], while commercial customised solutions for vehicle state estimation exist, such as the RT3002 [27], these may be unavailable in prototype projects with a restricted budget or for use in series production. Another problem with the custom made solutions might be in the device flexibility - the results may become unreliable when the driving conditions change.
It has been demonstrated by researchers in [6] that a combination of GPS and inertial sensors can provide accurate measurements of the vehicle side-slip and tyre slip angles. This is why the vehicle velocity v and its yaw rate $\dot{\psi}$ are

used as the main measurement quantities. Methods, that design observers to estimate the side-slip, often depend on accurate tyre parametrization, which might be problematic, since these vary based on the road surface [17]. The advantage of our approach is that some of the tyre parameters are estimated, creating a flexibility for changes of the road surface during the ride.

1.2 Goals of the thesis

There are two main goals of the thesis:

- Filtration of vehicle measured states: its velocity, yaw rate and estimation of a side-slip angle
- Estimation of tyre variables: the wheel load and peak factor

First part uses modelling of the vehicle to provide an accurate description of the vehicle, using Kalman filter for the estimation, which is compared to measured data. The measurements are data obtained from an electric formula.
Second part uses the available measurements (estimates) from the first part to estimate tyre variables. The main focus is on the vehicle side-slip angle and variables from Pacejka's magic formula, describing the forces acting on the tyre, the peak factor and wheel load combined. Using these estimates it is further possible to get the estimate of a tyre slip angle.

1.3 Outline

The main topic of the thesis is Kalman filtering and estimation of vehicle states. The work is divided into several parts. At the beginning, *Ch.2: System modelling* describes the modelling approach used for the description of the vehicle. Later on, *Ch.3: Data generation* is walking through the process of data generation, what conditions have been used, the setup for the data and also the post-processing of measured values. *Ch.4: Kalman filtering* is serving as a basis for the Kalman filter algorithm, describes how it works, its possible versions and the goals, which we want to achieve using it. *Ch.5: Experiments* presents the results of both virtual experiments demonstrating the filtering abilities and comparing the linearised model with the created non-linear one. The validation on the real measured data follows, with the verification of created model and estimation of values being presented in the process. *Ch.6: Performance evaluation* identifies additional methods for Kalman filter performance testing, using the error and some of the statistical tests. The tests are also used for tuning and setup of Kalman filter initial parameters, that process is described within the chapter. Last chapter, *Ch.7: Conclusions* summarizes the work done, comments on the results and suggests the future work related either by extension of the thesis topic or in further control design.

Chapter 2

System modelling

This chapter is adopted from the master's thesis from Denis Efremov [10]. Above all, we need to describe the system we are working with as accurately, as possible, to minimise the room for errors and improve the performance of a Kalman Filter (KF) . His thesis serves as a foundation for further work. Our goal focuses on the estimation of a single-track non-linear model (S-T) vehicle states and thus it uses the following derivations and assumptions.

- All lifting, rolling and pitching motion of the vehicle is neglected.
- Vehicle mass is assumed to be concentrated at the centre of gravity.
- Front and rear tyres are being represented as a single tyre at the centre of each axle.
- Imaginary contact points of tyres on the surface are assumed to lie along the centre of axles.
- Pneumatic trails and aligning torques resulting from a side-slip angles of tyres are neglected.
- Mass distribution on the axles is assumed to be constant.

A full model takes into account both dynamics and kinematics of the vehicle all the way down to tyres. As mentioned in the assumptions, we are using a representation of a single tyre on the front, respectively rear side of the vehicle. Therefore we have a separate input for each wheel.
We have three inputs for each wheel, resulting to six in total, first two being steering angles, δ_F, δ_R of the wheel, next pair is the driving torques τ_F, τ_R and the last pair is the braking torques τ_{BF}, τ_{BR}.

The system model is divided into several parts, which can be easily introduced using the following chart diagram. The inputs are marked in blue and the outputs in green. The states are highlighted in red :

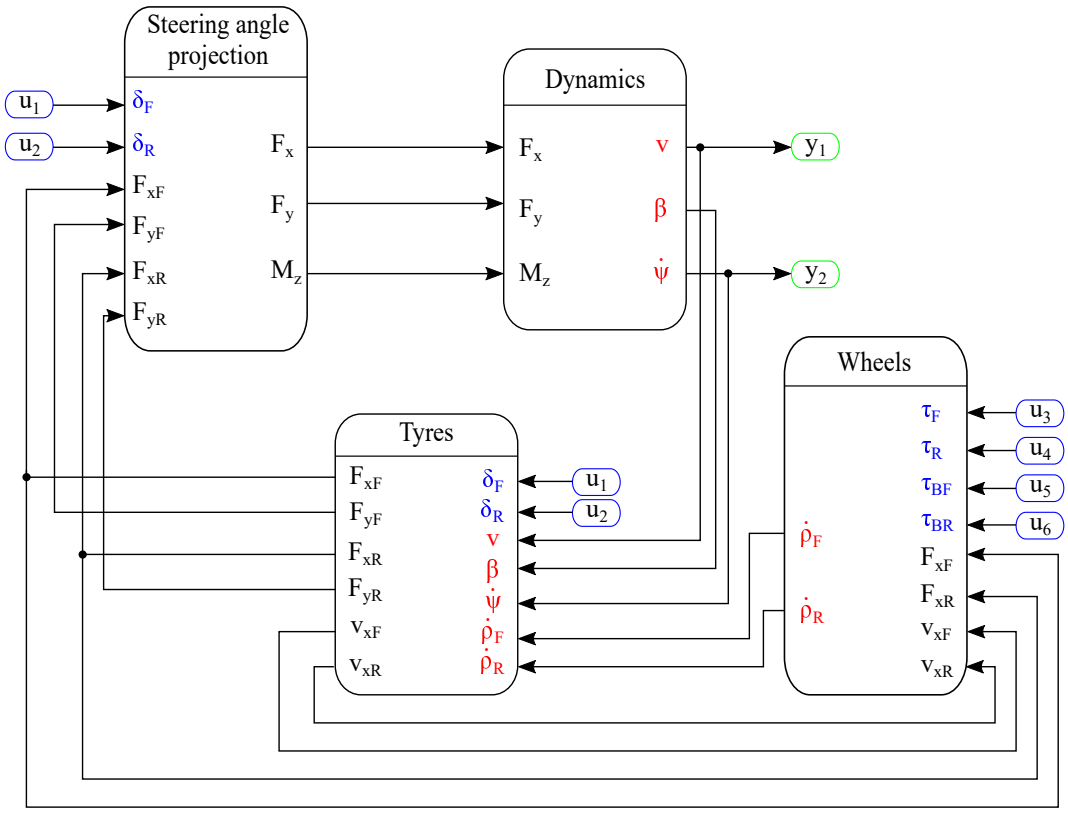


Figure 2.1: Non-linear model block diagram

2.1 Dynamics

The vehicle dynamics describe the behaviour of the vehicle after some of the driver inputs are performed - in our case we are talking about steering, acceleration and braking. The important system variables, which help us map these changes, are forward velocity v, side slip angle β and yaw rate $\dot{\psi}$.

These three variables have been selected as states of the system and using equations for S-T model alongside with the degrees of freedom movements we can make out some of the state-space equations.

2.1.1 Vehicle dynamics

An early assumption we have made is that the two tyre slip angles on the same axle (front, respectively rear) are the same. Under this assumption we can use the simplification of a one wheel at the centre of each axle. Every wheel is affected by longitudinal and lateral forces, F_x and F_y. Figure 2.2 shows all forces, angles and velocities used for modelling the vehicle dynamics.

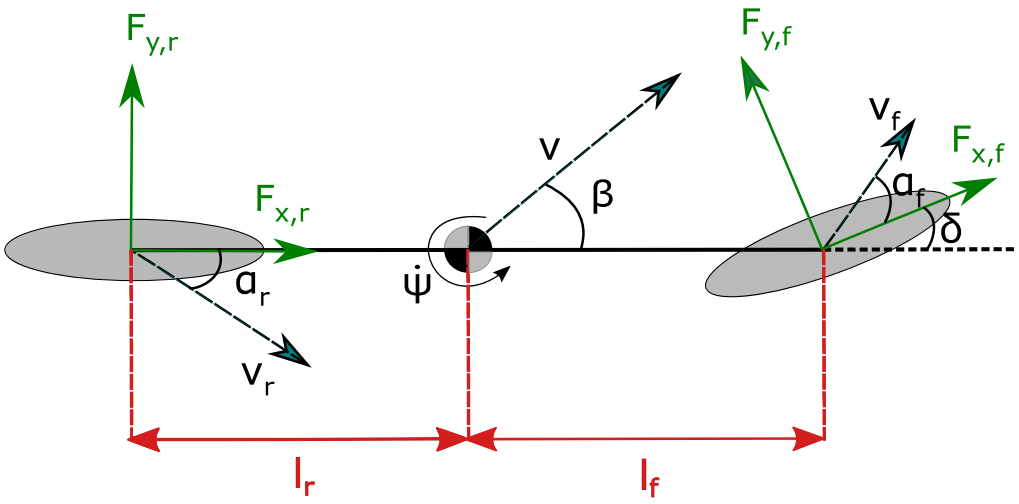


Figure 2.2: The single track model, adopted from [10]

We can see from figure 2.2 that in this system we have three degrees of freedom:

- **Longitudinal** motion, described by force F_x

$$F_x = m\dot{v}cos(\beta) - mv(\dot{\beta} + \dot{\psi})sin(\beta) \tag{2.1}$$

- **Lateral** motion, described by force F_y

$$F_y = m\dot{v}sin(\beta) - mv(\dot{\beta} + \dot{\psi})cos(\beta) \tag{2.2}$$

- **Vertical** (yaw) motion, from the moment around the z-axis, M_z

$$M_z = I_z\ddot{\psi} \tag{2.3}$$

Here m is a mass of the vehicle, $v, \dot{v}$ is the velocity, resp. acceleration of the vehicle's CoG, $\beta, \dot{\beta}$ is the side-slip angle, resp. angular velocity of the vehicle and I_z is the vehicle's moment of inertia around the z-axis.

From the equations above we can express the movements as functions of individual forces acting in each direction:

$$\begin{bmatrix} mv(\dot{\beta}+\dot{\psi}) \\ m\dot{v} \\ I_z\ddot{\psi} \end{bmatrix} = \begin{bmatrix} -sin(\beta) & cos(\beta) & 0 \\ cos(\beta) & sin(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} \quad . \tag{2.4}$$

Combining equations 2.6 and 2.4, we get the state space model description for the velocity v, side-slip angle β and yaw rate $\dot{\psi}$, which are the states of state equations:

$$\begin{aligned} \dot{v} &= \frac{1}{mv}\left[F_y cos(\beta) - F_x sin(\beta)\right] - \dot{\psi} \\ \dot{\beta} &= \frac{1}{m}\left[F_y sin(\beta) + F_x cos(\beta)\right] \\ \ddot{\psi} &= \frac{1}{I_z} M_z \quad . \end{aligned} \tag{2.5}$$

This set of equations describes the vehicle dynamics, and is also called a *Dynamics equation*. It incorporates the steering angle projection into the dynamics and provides a good insight into the vehicle's behaviour.

2.2 Kinematics

Every movement of a rigid body can be considered as a superposition of two movements: the translational movement of a fixed body point and the rotation of the body around this point [16]. In the kinematics block we are focusing mostly on wheels and tyres of the vehicle. These include highly non-linear relations, which make the estimation and Kalman filtering a challenging task. Since we are using a S-T model, all of the equations apply to this scenario.

Coordinate system

We are using a three dimensional Cartesian coordinate system with the x-axis pointing towards the observer for a description of our dynamics. The directions and rotations can be seen in figure 2.3. The x-axis describes a longitudinal motion, y-axis lateral, and the z-axis a lifting motion. The yaw rate $\dot{\psi}$ is positive, if the vehicle is turning to the left hand side.

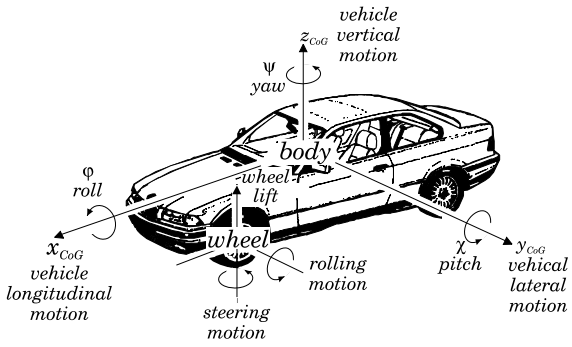


Figure 2.3: The vehicle coordinate system, adopted from [20]

The acting forces can be projected using the steering angles $\delta_{F,R}$:

$$\begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} = \begin{bmatrix} cos(\delta_F) & -sin(\delta_F) & cos(\delta_R) & -sin(\delta_R) \\ sin(\delta_F) & cos(\delta_F) & sin(\delta_R) & cos(\delta_R) \\ l_f sin(\delta_F) & l_f cos(\delta_F) & -l_r sin(\delta_R) & -l_r cos(\delta_R) \end{bmatrix} \begin{bmatrix} F_{xf} \\ F_{yf} \\ F_{xr} \\ F_{yr} \end{bmatrix} \quad . \tag{2.6}$$

Equation 2.6 is called a force coordinate transformation and it is a good way to show what effects do the front and rear steering angles $\delta_{F,R}$ have on the forces acting in each direction on the Centre of Gravity (CoG) of the vehicle.

2.3 Wheels modelling

For the description of wheels we are using the same coordinate system as for the whole model in 2.2, meaning that the x-axis is facing the observer. The state variable we have chosen for wheels is a rotation velocity of front, resp. rear wheel, $\dot{\rho}_{F/R}$, where the state equations show the rotation acceleration as follows [10]:

$$\begin{aligned} \ddot{\rho}_f &= \frac{1}{J_f}(\tau_F - R_f F_{xf} - sign(\dot{\rho}_f)\tau_{BR} - k_f v_{xf}) \\ \ddot{\rho}_r &= \frac{1}{J_r}(\tau_R - R_r F_{xr} - sign(\dot{\rho}_r)\tau_{BF} - k_r v_{xr}) \quad . \end{aligned} \tag{2.7}$$

Here the constants are radii of wheels $R_{f,r}$ and coefficients of the road drag for each wheel $k_{f,r}$. The inputs are the driving torques $\tau_{F,R}$ and braking torques $\tau_{B_{F,R}}$. Other variables in the equation are moments of inertia for each wheel $J_{f,r}$, forces acting on the centre of each wheel along the x-axis (from wheel rolling motion) $F_{xf,r}$ and the wheel travel velocity $v_{xf,r}$, defined in eq. 2.9 A free rolling wheel can be identified by its circumferential velocity[28]:

$$v_{ci} = \dot{\rho}_i R_i \quad . \tag{2.8}$$

The travel velocity differs in direction to the circumferential velocity by a tyre slip-angle α. This can be seen in fig. 2.3 and the relation between these two velocities is used to calculate the slip ratio λ. Travel velocity each wheel (front,rear) is defined as [10]:

$$\begin{aligned} v_{xf} &= vcos(\beta)cos(\delta_f) + (vsin(\beta) + l_f\dot{\psi})sin(\delta_f) \\ v_{xr} &= vcos(\beta)cos(\delta_r) + (vsin(\beta) - l_r\dot{\psi})sin(\delta_r) \quad . \end{aligned} \tag{2.9}$$

This velocity can be obtained from the odometry measurement.

An important relation describing the tyre longitudinal movement is between the longitudinal force F_x acting on the wheel and its slip ratio λ. The slip ratio has more alternative definitions, but we are using the SAE J670 definition, for tyres pointing straight ahead from [5],[25]:

$$\lambda_i = \frac{v_{ci} - v_{xi}}{v_{xi}} \quad , \tag{2.10}$$

where $-1 \leq \lambda \leq 1$, the ratio is negative, when braking and positive when driving.

2.3.1 Tyre modelling

Tyre characteristics are of crucial importance for the dynamic behaviour of road vehicle [28]. The main parameters we are working with are the tyres' slip angles $\alpha_{F,R}$, which show us an angle between the wheel plane and the velocity of the wheel ground contact point [20]. These angles are important for the process of estimation, because they are dependent on the steering angles $\delta_{F,R}$, and the states: side-slip angle β, velocity of the vehicle's centre of gravity v and the yaw rate $\dot{\psi}$. These variables except for the side-slip angle can be easily measured by the combination of GPS and on-board inertial sensors. Once we get the estimate of vehicle slip angles, we could use the tyre slip angles for further modelling and estimation of the tyre lateral forces models which are described later in this chapter.

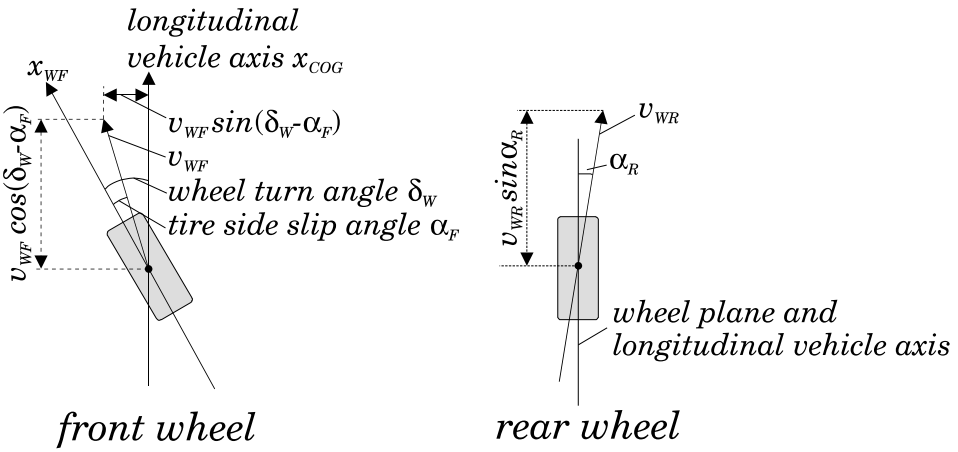


Figure 2.4: Tyre slip angles, adopted from [20]

Using the figure 2.4 and the equations in [20] we can express the tyres' slip angles as:

$$\alpha_F = -arctan\frac{[vsin(\beta) + l_f\dot{\psi}]cos(\delta_F) - vcos(\beta)sin(\delta_F)}{|[vsin(\beta) + l_f\dot{\psi}]sin(\delta_F) + vcos(\beta)cos(\delta_F)|}$$
$$\alpha_R = -arctan\frac{[vsin(\beta) - l_r\dot{\psi}]cos(\delta_R) - vcos(\beta)sin(\delta_R)}{|[vsin(\beta) - l_r\dot{\psi}]sin(\delta_R) + vcos(\beta)cos(\delta_R)|} \quad , \tag{2.11}$$

where $l_{F,R}$ are distances from the vehicle's centre of gravity to the front, resp. rear wheels.

In order for a tyre to produce a force, a slip must occur [30]. The knowledge of tyres' lateral and longitudinal forces allows us to get certain variable parameters of the car, like its wheel loads F_z, which vary on different roads and also during driving manoeuvres. The wheel load can simply be pre-computed as a constant, however, the next estimations could not be accurate for the full duration of the ride.

2.3.2 Friction ellipse

The friction ellipse (or for specific cases called the *Kamm's circle*) shows us how much of the force is being transferred from the tyre to the road surface. The combination of longitudinal and lateral forces acting on the wheel cannot be greater than the wheel load F_z. These combined slips occur during the acceleration or braking situations. If the geometric sum of longitudinal and lateral wheel forces lies within the ellipse, the resultant tyre forces can be transmitted to the ground [26]. We can see the borders in figure 2.5, where F_x grows during acceleration and diminishes while braking.

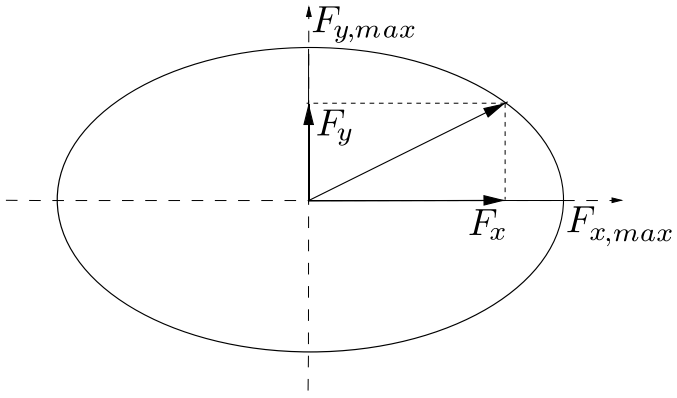


Figure 2.5: The friction ellipse

The combined force from a friction ellipse is defined as:

$$F = \sqrt{aF_x^2 + bF_y^2} \leq \mu F_z \quad , \tag{2.12}$$

where a, b are parameters of the ellipse and μ denotes the friction coefficient of a road (examples from [17]: pavement $\mu = 1$, gravel $\mu = 0.6$).

There are several models which can be used for modelling of the tyre forces. These are well compared by Lukas Haffner in [14]. We will compare only two of them, the very famous Pacejka's Magic formula (PMF) and its linear two-line approximation. Both models depend on tyre slip angle α, which was explained and defined on the previous page.

2.3.3 Pacejka's tyre model

The modelling approach allows that the characteristics of the lateral guiding forces, the braking forces and the aligning torque are described mathematically. Used formulas are capable of describing the circumferential force, lateral force and aligning torque as functions of the longitudinal and lateral slip with a high degree of accuracy [31]. Wheel forces for each wheel (i=front,rear) can be obtained through following formula from Pacejka [28]

$$\begin{aligned} F_{yi} &= D_y F_{zi} sin\left[C_y arctan(B_y\alpha_i - E_y(B_y\alpha_i - arctan(B_y\alpha_i)))\right] \\ F_{xi} &= D_x F_{zi} sin\left[C_x arctan(B_x\lambda_i - E_x(B_x\lambda_i - arctan(B_x\lambda_i)))\right] \quad , \end{aligned} \tag{2.13}$$

where in the lateral force we are using the tyre slip angle α and in the longitudinal case the tyre slip ratio λ, the coefficient B is a stiffness factor, C a cornering stiffness, D is a peak factor including the tyre friction coefficient μ which varies on different road surfaces and E is a curvature factor. All of these coefficients can be expressed as functions of the nominal wheel load F_z and are different for wheel lateral forces F_y and F_x.
The equation 2.13 is called a Simplified Pacejka Magic formula, because it is a lighter version of the original PMF, which includes many more coefficients. For our application this simplified formula is more than satisfying, there is therefore no need to use the full PMF.
As we can see from eq. 2.13, this description is highly non-linear, which might cause problems with further use, especially in the region around $\lambda = 0$, or very small tyre slip angles α - see fig. 2.5. This is why the knowledge of coefficients B, C, D, E is very useful and allows us using this accurate tyre model. They are, however, not always known. Depending on the tyre rubber materials, it is mostly a secret of the tyre manufacturing companies. In case we do not know these, we could use the less accurate, but very similar and linear two-line tyre model.

2.3.4 Two-line tyre model

The main purpose of a two-line tyre model is to get as close to Pacejka's approximation as possible with not as many coefficients. It consists of two saturations and a linear function for the most problematic areas, around $\lambda = 0$, resp. $\alpha = 0$. The linear part is a first-order approximation of the tyre force characteristics and it is obtained from Taylor expansion of the tyre cornering force characteristics [11]:

$$C_\alpha = \left.\frac{\partial F_y}{\partial \alpha}\right|_{\alpha=0} \quad . \tag{2.14}$$

This constant is the same as the cornering stiffness C_y in eq.2.13. The two-lines model (or sometimes called a *Linear bicycle model of lateral vehicle dynamics* [25]) is defined for lateral case as [10]:

$$F_y = \begin{cases} C_\alpha \alpha_i & |\alpha_i| < \frac{\mu_{max}}{C_\alpha}, \\ \mu_{max} & |\alpha_i| \geq \frac{\mu_{max}}{C_\alpha} \end{cases} \quad . \tag{2.15}$$

where μ_{max} is a maximal friction coefficient depending on the road surface. Similar relations would apply in the longitudinal case, where in equations 2.14 and 2.5 we would change the tyre slip angle α for the slip ratio λ and the lateral force F_y for longitudinal F_x. That gives us different coefficient C_λ for eq. 2.5.
In our non-linear modelling we are using the Pacejka's Magic formula because of the accuracy, the Two-line tyre model is used for linearised model only for its simplicity. Both models can be seen in fig.2.6

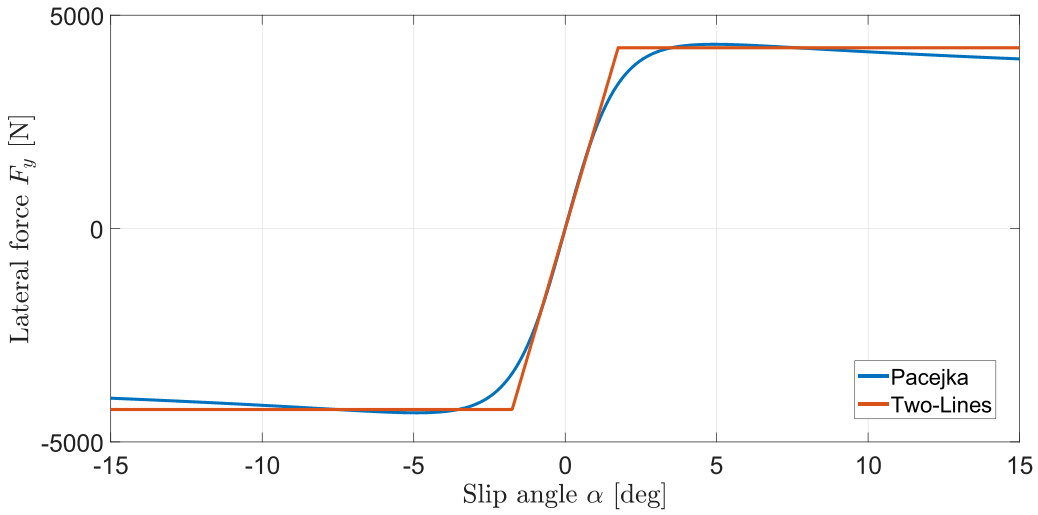


Figure 2.6: Comparison between Pacejka's and Two-line tyre model, adopted from[10]

2.4 Modelling outputs

The modelling done above gives us several outputs. Firstly, we have a set of non-linear equations describing the model we are using for our data validation in 3.1. This model shows us the accuracy of our approach and estimations when set for a particular vehicle (very different configurations for normal cars like Škoda, Porsche and for Formula) - with the weights, moments and dimensions of the real vehicle. The incorrect setup might lead to vehicle instability.
Secondly, we can also get a linearised model using various assumptions. This model is derived in chapter 3.2 and can be useful for comparison between a linear approach and fully non-linear modelling. Another use of linearised model can be for controllers design, where a simplified model description might be sufficient. This model can also help avoiding the problematic non-linear areas with an operating point chosen correctly.

Chapter 3

Data generation

This chapter describes the process and options for data generation. Firstly, the individual car constants are loaded. There are several different car configurations available to choose from: Škoda Fabia, BMW, Porsche 911, Porsche 928 and *Formula*.
Secondly, we also obtained data from the eForce FEE Prague Formula[1] . This set of data is interesting from the point of view where we validate the created algorithms, containing all the noises and other unexpected scenarios, testing the KF robustness. On the other hand it is also rather challenging task using this set of data, because of the fast changes of the vehicle states during high-speed manoeuvres. We also got the dimensions and other properties of the formula, thus we were able to create an additional configuration simulating the formula behaviour in the option *Formula*.
The utilized models correspond to the vehicle kinematics and dynamics equations described in chapter 2: System modelling. We have created two scenarios to validate and demonstrate the vehicle behaviour:

- Steady straight ride with a constant velocity
- Steady turning ride with constant turning rate and velocity

Steady straight ride is useful for the simulation where we want to observe the raw behaviour of the vehicle without any additional forces distorting the measured signals. On the other hand, the steady turning ride can be useful for testing the robustness of the algorithm, since it adds more complicated characteristics to work with and challenges the estimation in an additional way - for some inputs the vehicle becomes unstable.
The simulations performed have taken into account various inputs characteristics starting from a doublet all the way to some sine-wave inputs on the steering wheel imitating the steer from one side to another with no steady periods. Distinct torque acceleration and braking inputs have also been tried out.

[1] eForce FEE Prague Formula - https://eforce.cvut.cz/

Because of the KF usage, we added noise to the measured data as well. The state noise can symbolise some fault inside the vehicle (for instance a tyre defect) and the measurement noise can be of a faulty or inaccurate measurements. For further testing of the KF algorithm, we used different levels of the signal-to-noise ratio (SNR) on the measured data. Furthermore, the states of the vehicle have been also subjected to the noise in order to get a fully stochastic system.

3.1 Non-linear model

The non-linear model uses the modelling approach from chapter 2: System modelling for as accurate description of the vehicle as possible. The block representation of the model is provided in fig. 2.1
The input vector u is formed by steering angles $\delta_{F,R}$, wheel torques $\tau_{F,R}$ and braking torques $\tau_{BF,R}$:

$$u = \begin{bmatrix} \delta_F & [rad] \\ \delta_R & [rad] \\ \tau_F & [Nm] \\ \tau_R & [Nm] \\ \tau_{BF} & [Nm] \\ \tau_{BR} & [Nm] \end{bmatrix} \quad . \tag{3.1}$$

The state non-linear representation we are using contains five states: the vehicle forward velocity v, side-slip angle β, yaw rate $\dot{\psi}$ and front wheel rotation velocities $\dot{\rho}_{f,r}$. Combining the equations describing wheels 2.7 and the dynamics equations 2.5 we get the state non-linear representation:

$$\begin{aligned} \dot{v} &= \frac{1}{mv}\left[F_y cos(\beta) - F_x sin(\beta)\right] - \dot{\psi} \\ \dot{\beta} &= \frac{1}{m}\left[F_y sin(\beta) + F_x cos(\beta)\right] \\ \ddot{\psi} &= \frac{1}{I_z} M_z \\ \ddot{\rho}_f &= \frac{1}{J_f}(\tau_F - R_f F_{xf} - sign(\dot{\rho}_f)\tau_{BR} - k_f v_{xf}) \\ \ddot{\rho}_r &= \frac{1}{J_r}(\tau_R - R_r F_{xr} - sign(\dot{\rho}_r)\tau_{BF} - k_r v_{xr}) \end{aligned} \quad , \tag{3.2}$$

where the moment of the vehicle around z-axis is:

$$M_z = F_{yf} l_f cos(\delta_F) - F_{yr} l_r cos(\delta_R) + F_{xf} l_f sin(\delta_F) + F_{xr} l_r sin(\delta_R) \quad . \tag{3.3}$$

The *basic* state vector therefore is:

$$x = \begin{bmatrix} v \ [ms^{-1}] \\ \beta \ [rad] \\ \dot{\psi} \ [rads^{-1}] \\ \dot{\rho}_f \ [rads^{-1}] \\ \dot{\rho}_r \ [rads^{-1}] \end{bmatrix} \quad . \tag{3.4}$$

The word *basic* has been used on purpose, because in the estimation part we would also like to estimate the coefficient DF_z in the Pacejka's magic formula (2.13), in that case we add two artificial states to the state vector, D_xF_{zf}, resp. DF_{zr}.
Finally, the output measurement vector does only contain the velocity v and yaw rate of the vehicle $\dot{\psi}$, because both of these states can easily be measured (using the on-board odometry and gyroscopic sensor) for estimation of the side-slip angle β, in the case of DF_z a sum of rotational velocities $\dot{\rho}$ has been added.

$$y = \begin{bmatrix} v \ [ms^{-1}] \\ \dot{\psi} \ [rads^{-1}] \end{bmatrix} \quad . \tag{3.5}$$

3.2 Linearised model

For linearisation of the non-linear model additional assumptions have been made [10]:

- The steering angles $\delta_{F,R}$ are smaller than 10°
- Side-slip angle of the vehicle β is smaller than 10°

These assumptions result into a general simplification:

$$\begin{aligned} sin(x) &\approx x \\ cos(x) &\approx 1 \quad . \end{aligned} \tag{3.6}$$

Applying the linearisation of a S-T model from [2] and using all the simplifications we will get the eq.2.4 in a form of:

$$\begin{bmatrix} mv(\dot{\beta} + \dot{\psi}) \\ m\dot{v} \\ I_z\ddot{\psi} \end{bmatrix} = \begin{bmatrix} -\beta & 1 & 0 \\ 1 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} \quad . \tag{3.7}$$

Also, at the beginning of this chapter there are two scenarios listed. Both are working with a constant travelling velocity v. From this design we can simplify $\dot{v} = 0$. Using this simplification we will get in the second row of eq.3.7: $F_x = -\beta F_y$.

Furthermore, the vehicle side-slip angle β is small, with a property from [2]: $\beta^2 \ll 1$. We can simplify eq.3.7 to:

$$\begin{bmatrix} mv(\dot{\beta} + \dot{\psi}) \\ I_z\ddot{\psi} \end{bmatrix} = \begin{bmatrix} F_y \\ M_z \end{bmatrix} \quad . \tag{3.8}$$

In the perspective of tyres, the slip angles are simplifying eq. 2.11 into [14]:

$$\begin{aligned} \alpha_F &= \beta + \frac{l_f\dot{\psi}}{v} - \delta_F \\ \alpha_R &= \beta - \frac{l_r\dot{\psi}}{v} - \delta_R \quad . \end{aligned} \tag{3.9}$$

Following this simplification and taking into account the second assumption, that the steering angles $\delta_{F,R}$ are small, we can now use the Two-line tyre model as a linear approximation around $\alpha = 0$. In this region, the lateral forces from eq. 2.13 are now the approximated through the first part of 2.5:

$$\begin{aligned} F_{yf} &= C_{\alpha_F}\alpha_F = C_{\alpha_F}\left(\beta + \frac{l_f\dot{\psi}}{v} - \delta_F\right) \\ F_{yr} &= C_{\alpha_R}\alpha_R = C_{\alpha_R}\left(\beta - \frac{l_r\dot{\psi}}{v} - \delta_R\right) \quad . \end{aligned} \tag{3.10}$$

The total lateral force acting on the wheel F_y is from the eq. 2.6 following:

$$F_y = F_{xf}sin(\delta_F) + F_{yf}cos(\delta_F) + F_{xr}sin(\delta_R) + F_{yr}cos(\delta_R) \quad . \tag{3.11}$$

But applying the assumptions from the beginning of this chapter and also 3.6, the longitudinal parts can be omitted ($F_{xf}sin(\delta_F) \approx F_{xf}\delta_F \approx 0$, resp. $F_{xr}sin(\delta_r) \approx F_{xr}\delta_r \approx 0$). The omission corresponds with the general assumption that the car is travelling with small steering angles. Lateral force is then altered to:

$$F_y = F_{yf} + F_{yr} \quad . \tag{3.12}$$

Combining all the simplifications made due to linearisation, we can write the linearised state space representation as [2]:

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} \delta_F \\ \delta_R \end{bmatrix} \quad . \tag{3.13}$$

where the coefficients are:

$$\begin{aligned}
a_{11} &= -\frac{C_{\alpha_F} + C_{\alpha_R}}{mv} \\
a_{12} &= \frac{l_r C_{\alpha_R} - l_f C_{\alpha_F}}{mv^2} - 1 \\
a_{21} &= \frac{l_r C_{\alpha_R} - l_f C_{\alpha_F}}{I_z} \\
a_{22} &= -\frac{l_f^2 C_{\alpha_F} + l_r^2 C_{\alpha_R}}{I_z v} \\
b_{11} &= \frac{C_{\alpha_F}}{mv} \\
b_{12} &= \frac{C_{\alpha_R}}{mv} \\
b_{21} &= \frac{l_f C_{\alpha_F}}{I_z} \\
b_{22} &= -\frac{l_r C_{\alpha_R}}{I_z} \qquad .
\end{aligned} \tag{3.14}$$

3.3 The eForce formula

The last set of data we worked with is from the electric racing student formula (shown in fig. 3.1), which we were given by a colleague Ing. Marek Laszlo, who is also a driver of the formula. Some of the data like mass and inertia tensors with a display of CoG in the coordinate system can be seen in his Master's thesis [22]. An undisputed advantage in working with this set of measurements is the fact that it was obtained from a real vehicle, not through a simulation output in ideal conditions. We can therefore try to compare the created non-linear model from 3.1 with the behaviour of formula after defining its parameters, or simply try to test the EKF algorithm on measured data.

The formula is shorter in length compared to classic cars (the distance between front and rear wheels is about 1.5m, Škoda Fabia[2] has the wheelbase of almost 2.5m) and the biggest difference is in weights of various components resulting into much smaller total mass of the vehicle. Due to the relatively heavy Škoda (mass cca.1500 kg) we have not taken into account the driver's weight, since it represents only about 5% of the total mass. In the case of a formula, the heaviest part of the vehicle is the driver, and the ratio is significantly different, of about 1/3 of a total weight. For the illustration of this substantial assumption, the sum of the driver's and accumulator weight is more than 50% of the total formula mass.

Figure 3.1: The FEE eForce Formula, adopted from [9]

[2]data from Škoda Fabia owner's manual [23]

Due to the data being measured by the Internal measurement unit (IMU) are not always in a form of what is needed for further calculations, but are tailored as a feedback for the driver, we need to make the transformations or simplifications in a full compliance with the S-T model assumptions and used modelling approach. These include specifically:

- Rotation from NED coordinates (North-east-down)
- Averaging of wheels
- Computation of driving and braking torques
- Unit conversion (angular, torque)

The measured velocities are in the NED coordinates. This coordinate system is widely used in aviation and flight control. As we can see in fig.3.2, the body (index b) and the vehicle carried NED coordinates (index nv) are two Cartesian frames rotated from one another [4]. Using the Euler rotation angles sequence "*Yaw-Pitch-Roll*" we can convert the NED coordinates to the body frame coordinate system. This is necessary in order to use the equations from modelling chapter.

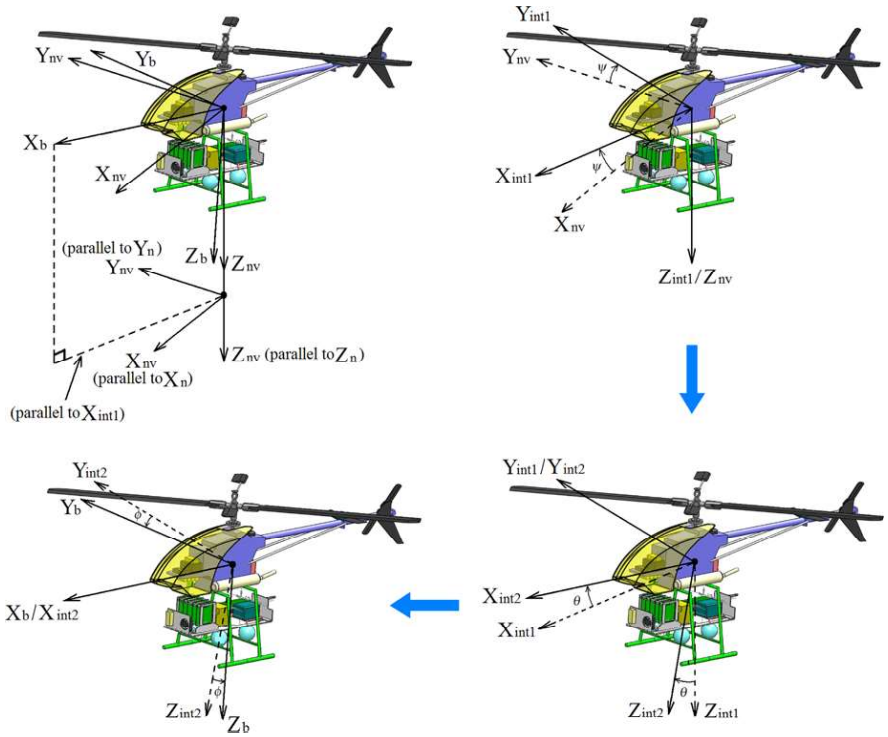


Figure 3.2: The NED coordinate system and Euler rotations, adopted from [4]

The first rotation, around the z-axis, is described by a yaw angle ψ. The rotation matrix between the local (nv) coordinates and intermediate (int) is as follows:

$$R_{int1/nv} = \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad . \tag{3.15}$$

Second rotation is around the y-axis, the pitch rotation, using the pitch angle θ:

$$R_{int2/int1} = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix} \quad . \tag{3.16}$$

And the last rotation is performed with a fixed x-axis using the roll angle ϕ.

$$R_{b/int2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix} \quad . \tag{3.17}$$

The total rotation and coordinate transformation are then:

$$\begin{aligned} R_{b/nv} &= R_{int1/nv} R_{int2/int1} R_{b/int2} \\ \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} &= \begin{bmatrix} x_{nv} \\ y_{nv} \\ z_{nv} \end{bmatrix} R_{b/nv} \quad . \end{aligned} \tag{3.18}$$

Using this equation in a similar way we can substitute velocities for coordinates and get its vectors for each direction.

The second possible option to obtain the vehicle velocity might be using the particular wheel speeds, which are also available from the measured data. Firstly, we have to do the average wheel speed since we are using the S-T model, which assumes a single tyre at the centre of each axis, therefore we merge the two front (resp. rear) wheels into one on the particular axle with an average speed. This method can be used for a comparison of results using different approaches.
Furthermore, since two of the states are the wheel angular velocities $\dot{\rho}_{F/R}$, which can be computed just by multiplication (eq. 2.10: $\dot{\rho} = v_{ci} R$) using the formula's wheel radius ($R = 0.195\ m$) and a wheel velocity v_{ci}, we can also observe the wheels' angular velocity to make the output vector more detailed for better KF error comparison.

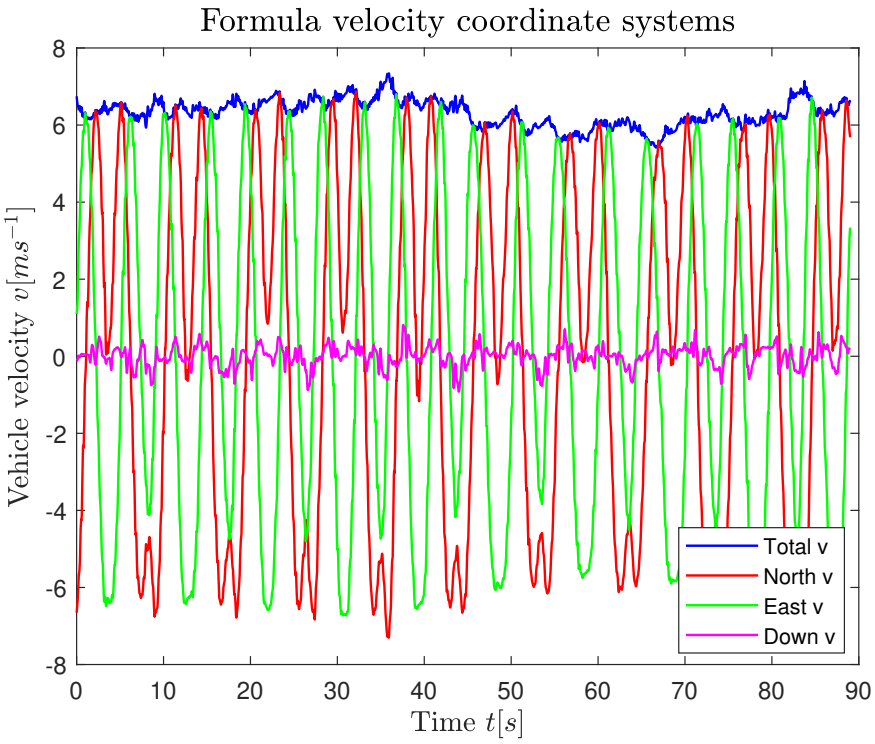


Figure 3.3: The formula velocities from two coordinate systems comparison

For a better practical use, the formula's motor performance is measured using the revolutions per minute (RPM). However, in our model we are using driving/braking torques, which have to be in Nm. For the knowledge of the motor torque we still need to get the power of the motor.
In the formula, the induction motor voltage and current is measured using the direct and quadrature (d-q) axis model, which is used for easier control applications. By having the voltage and current quantities in d-q frame, it is possible to control the speed of the machine by controlling the flux and torque independently [29].
Using the motor RPM request signal and its power, we are able to get the torque generated. Using the rotational velocity $\omega = 2\pi/60$ $[rads^{-1}]$ we are able to use the RPM in an equation:

$$\tau_M = \frac{P}{\omega RPM} \quad . \tag{3.19}$$

The wheel driving torque goes from the motor through the transmission with a particular ratio (TR) and it can be computed as:

$$\tau_W = \tau_M TR \quad . \tag{3.20}$$

Similarly, there is a ratio between the the steering wheel turning angle and the real turn performed on the tyres. This ratio is about 3.8, meaning that if the steering wheel is turned by 95°, the wheels turn by 25°. The conversion was needed, because the measured signal was the steering wheel angle, which we do not need precisely, since we are interested in tyres' steering angles.

The process of getting braking torques was a bit more complicated. The measured data from brakes was only in braking pressure (kPa). Since we did not exactly know the size of the braking pads (from which we would use the braking area) and their friction coefficient, we were not able to compute precisely the braking torque. Based on the experienced estimate from the driver, the braking pressure was converted into torque linearly, with the basic approximation of $p = 3000kPa \approx 800Nm$ of braking torque and shifted by a constant offset for the braking torque to reach around zero when the car is not using brakes (the use of brakes was indicated in a separate binary signal measuring if the braking pedal has been stepped on).

Chapter 4

Kalman filtering

Every real, measured signal is exposed to some level of a noise. These systems, which are affected not only by a given deterministic input, but also with other, random processes, are called *stochastic* systems [15]. This addition of a random quantity lead to the system states and outputs being also random processes. The source of a disturbance can be external, from another device causing interference via electromagnetic radiation or simply due to incorrect measurement from the sensor, resulting into a noise, which is distorting the results. This type of a noise is called a *measurement noise*.
Another type of the noise comes from the inside of a system. When the states of a system change over time, with unknown exact details of when/how those changes occur, we need to model them as a random process. An example of this change can be illustrated as a defect within the system itself providing inaccurate measurements. For instance, in our case of a vehicle system, we can have a damaged tyre, which supposedly with no damage is rotating with a known rotational velocity. Even with a correct and precise measurement of wheel rotation we will still be getting different (wrong) values of the velocity from the known ones - the tyre dynamics are perturbed and therefore the system does not behave as expected. This difference is caused by a *process noise*.

In many dynamic mechanical systems, it is often unrealistic to assume that all states describing the system's behaviour can be measured [7]. In some cases, the states are not accessible for sensors or they simply cannot be measured. This provides us with a problem where estimation and filtering can become very useful as a solution, using other, easily measurable signals to derivate a particular state value, which can be used for control design. The real challenge is to provide accurate state estimates without decreasing its accuracy or the algorithm robustness.
The Kalman filter is a powerful tool for analysis and the use of stochastic system measurements. It allows us to filter the noisy signals sometimes even through a very noisy signal when set correctly and also estimate the states we are not able to measure.

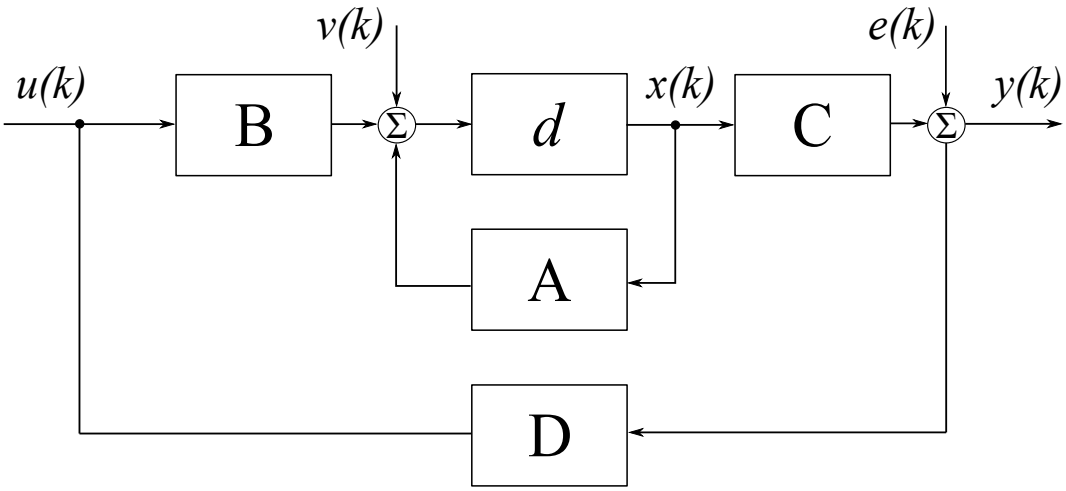


Figure 4.1: Linear stochastic system

Fig. 4.1 shows a structure of a discrete linear stochastic system with a time delay d, state-space description matrices A, B, C, D and both types of noises mentioned earlier. The measurement noise, $e(k)$, is added to the output of the system, the process noise is labelled as $v(k)$ and we can see that it is added to the state equation.
The noises can be thought of as random variables defined by their mean value and covariance [15]:

$$\begin{aligned} \varepsilon\left\{\begin{bmatrix} v(k) \\ e(k) \end{bmatrix}\right\} &= 0 \\ cov\left\{\begin{bmatrix} v(k_1) \\ e(k_1) \end{bmatrix}, \begin{bmatrix} v(k_2) \\ e(k_2) \end{bmatrix}\right\} &= \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta(k_1 - k_2) \quad . \end{aligned} \tag{4.1}$$

Our used assumptions for KF have been:

- Stable system state matrix A
- The system is fully observable
- Process noise and measurement noise are not correlated ($S = 0$) [15]
- Data-update and time-update step can be performed separately [15]
- The state estimate is a linear mean-square (LMS) estimate

4.1 Kalman filter abilities

Kalman filter can be used in several applications. The most known are:

- Estimating the states (or filtering, smoothing noisy values)
- Used as a whitening or noise shaping filter

Estimation or observation means the extraction of information of any physical variable not available from direct sensors, by using only available information [7]. This is mostly done with the use of Kalman gain in the KF computed using the system model, along with the setup of noise covariance matrices Q, R. The state estimate is found as an optimal LMS estimate minimising its optimality criterion.

Noisy signals are not welcome in the following signal processing. The measured signals can be distorted by different strengths of noise, making some of the measurements very hard to work with. That is why a KF is used, estimating the mean value of the noisy signal for a "clean" result of a measured values. The KF algorithm is quite powerful, working even with relatively large SNR, where the output signal does not show any usable data and the noise prevails. The application of a KF on a very noisy signal can be seen in the chapter 5.:Experiments

The noise entering the system can be of different properties - that is why it is divided into two main groups: the *white* noise and the *coloured* noise. These names have originated from the analogy with light spectra. If we get a spectrum of white light, all the frequencies are being present uniformly. On the other hand, with coloured light, some frequencies are dominant, which disturbs the homogeneity of the spectrum.
Same occurs in the frequency domain of noise spectra. White noise is a uniform mixture of energy being present at every frequency. The very important property of white noise is a zero expected mean value, which is used later in KF algorithm. Various colours of noise are specifically used in audio and video engineering. In order to get the coloured noise, a white noise has to be shaped. This can be done by a Kalman filter, working as a high/low-pass filter, with an output of a random process of particular spectral density (and consequently a desired colour of noise). On the contrary, if we have a coloured noise sequence, and need to use the stochastic system model, which assumes a white noise input, we add a Kalman filter as a whitening filter (its output is a white noise sequence).

4.2 Kalman filter structure

To start with a description, the best way is to note a similarity between the deterministic approach and the stochastic systems. In the deterministic case, we would have used a state observer. If the linear system is in an observable canonical form, by linearly injecting the output into the dynamics, we replace A by $(A - KC)$. If all the observer poles lie in the left half of the complex plane (in the continuous time domain), the error $\epsilon(t)$ is asymptotically stable, it approaches zero as time goes forward. The matrix K is then an observer gain matrix [24].
Similarly, in stochastic systems we are using this structure with a few differences. We can list first the most obvious one, the noise addition. In this case, we cannot be certain of the system behaviour, since the output of the system is a stochastic process too. The only deterministic part of the system is the input. Based on the input, system model and the measured output we are able to *estimate* the system state in the next time step (in the discrete time). Kalman filter then uses the conditional probability density function and mean square (MS) estimate to find the state value in the next time sample.
The KF structure can be seen in fig. 4.2. The error $\epsilon(k)$ is obtained as a difference between measured output of a stochastic system $y(k)$ and the estimated output $\hat{y}(k)$ from the KF. This is then multiplied by Kalman gain L, which is different in each step depending on the error size, and used to get the next state estimate $\hat{x}(k)$ in the KF.

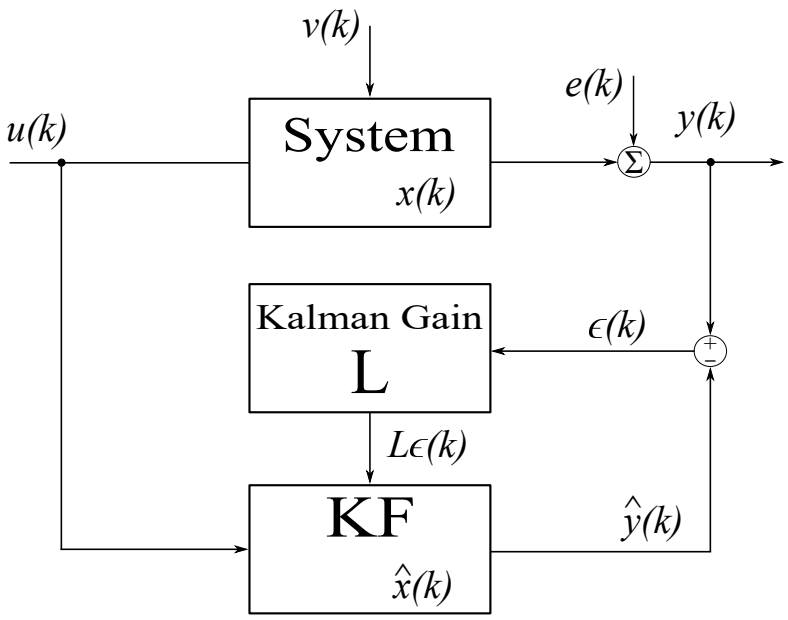


Figure 4.2: Kalman filter structure diagram

Because of the added noise (and therefore also some uncertainty in the system), we are using conditional probabilities - based on the measured output data and the known, deterministic input value. For notation purposes, $p(y|\mathcal{D}^{k-1})$ means a probability of the output y based on the data from time $(k-1)$.
With the intention of using the conditional probabilities, we need to know the distribution of our data. The joint conditional probability of state $x(t)$ and output $y(t)$ is from a normal distribution with a following mean value and since it is a vector, its covariance [15]:

$$p\left(\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} \middle| \mathcal{D}^{k-1}\right) = \mathcal{N}\left(\begin{bmatrix} \hat{x}(k|k-1) \\ \hat{y}(k|k-1) \end{bmatrix}; \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix}\right) \quad . \tag{4.2}$$

where $\hat{x}$ and $\hat{y}$ are the state, resp. output estimates.

The covariance matrix represents a relation between the variables, with the diagonal elements noting the variance of a particular variable, the off-diagonal elements contain the cross-covariances of each pair of variables. The aim of creating a covariance matrix is to show how large the changes are in data of the dataset (variance tells us how much the data are scattered around the mean value).

4.3 Noise covariance matrices

A correct initialisation of noise covariance matrices Q, R, which are describing the noise covariance in eq. 4.1 is the main tuning factor of the KF performance. Wrong setup of these covariance matrices might result into a failure and the performance of KF would be more harmful than beneficial. While the specification of the measurement noise covariance matrix, R, can be directly derived from the accuracy of characteristics of the measurement device, specification of the process noise covariance matrix Q is often attempted in a trial and-error approach [34]. In this work, noise covariance matrices Q, R are initialised as a diagonal matrices, with Q having different values of "weights" for each state.

Tuning the Kalman filter involves the selection of the process noise covariance matrix, Q. "If this matrix is guessed low, the filter will believe the model excessively and will not use the on-line measurements properly to correct the states. This can lead to poor performance or even filter divergence. On the other hand, if the matrix Q is guessed higher than the actual value, the state estimates will be noisy and uncertain, as this would lead to increased values of the state covariance matrix, P" [34].

4.4 Kalman filter algorithm

The algorithm itself is divided into parts, which are visualised in fig. 4.4. It uses the *a priori* and *a posteriori* information with the relation to the time sampling. These two terms are visualised on a time axis, in fig. 4.3. The single iteration (or time window) is always divided into two parts. The first one (on the left hand side of the window) being an *a priori* information, or "before we process the data" of the sample with a same index. As an example can be $x_{(k|k-1)}$ at time k. The second part of the window is an *a posteriori* information, indicating the use of data for the particular time index. The noted example $x_{(k|k-1)}$ is then working with "old" data acquired in the time window $k-1$.
To conclude this double indexing, the notation with the index on the left always shows the time at which the data is being analysed and the index on the right side of the bar indicates the data sample being used.

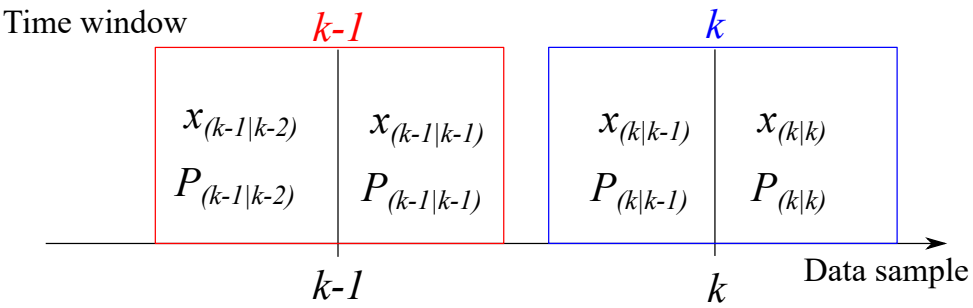


Figure 4.3: A priori and a posteriori data visualisation

Firstly, the algorithm has to be initialised with a priori data of state mean value estimate $\hat{x}_0$ and an initial state covariance P_0. This needs to be done, because we are using the data estimated for the next time window from a previous time iteration. After the initialisation a data-update step starts the estimation followed with the time-update step. Both steps are explained in detail in parts Data-update step and Time-update step.

The estimates are being performed using the LMS estimate. The advantage of LMS is, that it only requires means, variances and covariances of the data, whereas other estimators, like maximum likelihood need to know the joint probability distribution functions which are not easy to get.

Linear MS error estimators are easy to use, calculate, and are very versatile, since the resulting estimates are simply a linear function dependent on the output:

$$\hat{x}_{LMS}(y) = Ay + b \quad , \tag{4.3}$$

where the matrix A and vector b is chosen with the aim to minimise the following optimality criterion J_{LMS}:

$$J_{LMS} = \varepsilon \left\{ (x - \hat{x}_{LMS}(y))^T (x - \hat{x}_{LMS}(y)) \right\} \quad . \tag{4.4}$$

$\hat{x}_{LMS}$ is an LMS state estimate. The equation 4.3 can be rewritten as:

$$\hat{x}_{LMS}(y) = \mu_x + P_{xy} P_{yy}^{-1} (y - \mu_y) \quad . \tag{4.5}$$

using the mean values of a state and output $\mu_{x/y}$ along with the conditional state covariance matrix elements P_{xy}, P_{yy}.

The equations described in the next part form the basis of the Kalman filter, because [32]:

- The mean of the state is the KF estimate of the state
- The covariance of the state is the covariance of the KF state estimate

4.4.1 Data-update step

The KF has two different steps, which are being executed during a single iteration. The aim of *data-update* step (sometimes also called a *measurement* update [1]) is to use the newly acquired data (input $u(k)$ and output $y(k)$) to find the error $\epsilon(k)$ between an estimated output value $\hat{y}(k|k)$ and real measurement output $y(k)$. Then it is used to find the Kalman gain $L(k)$ for a particular data-update step. Finding the gain requires state P and measurement noise covariance matrix R , which are crucial for the KF algorithm, because they set up the accuracy of KF, which "corrects" the estimate done from a previous iteration. The Kalman gain is obtained as follows:

$$L(k) = P(k|k-1) C^T (C P(k|k-1) C^T + R)^{-1} \quad . \quad , \tag{4.6}$$

where C is the system output matrix. In the case of non-linear systems (and the use of EKF), the matrix C is different every time iteration, because it is a Jacobian of the set of output equations differentiated with respect to the state vector x seen in eq.(4.14). This basic structure of this process is described in fig. 4.2.

After obtaining the Kalman gain, the estimate, which has been predicted in a previous time step $\hat{x}(k|k-1)$, has to be updated in order to get an estimate for current time-sample iteration $\hat{x}(k|k)$. The new estimate is updated with the Kalman gain $L(k)$ and using the error ϵ we get it as:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + L(k)\epsilon(k) \quad . \tag{4.7}$$

The update of state covariance matrix takes into account the measurement noise covariance matrix R, acting as a tuning parameter with the Kalman gain L. The update for the state covariance matrix is produced in a following way:

$$P(k|k) = P(k|k-1) - L(k)(CP(k|k-1)C^T + R)L^T \quad . \tag{4.8}$$

After these updates, we can perform the time-update step, which is described in the following part.

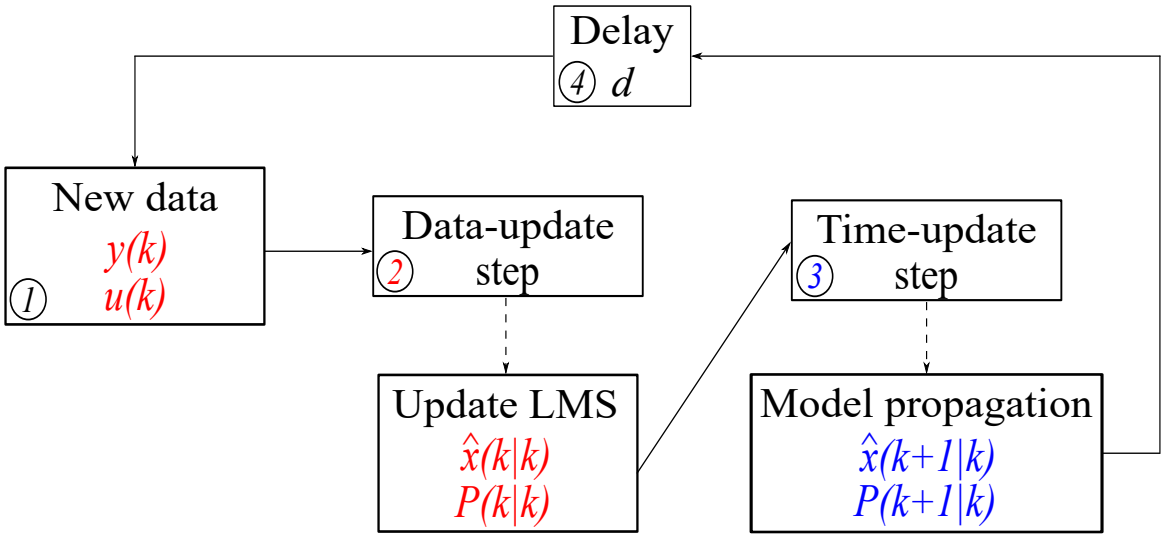


Figure 4.4: Data and time-update step

4.4.2 Time-update step

The main purpose of time-update step is to follow the state equations to get the value of the state in the next time iteration. It is done by using the estimate obtained for a current time window in the data-update step, $x(k|k)$, in the output equation. That way we can get the estimated output.

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) \quad . \tag{4.9}$$

Eq. 4.9 is called the time update equation for the estimate $\hat{x}$. From time k to time $k+1$, the state estimate propagates the same way that the mean of the state propagates. This makes sense intuitively. We do not have any additional measurements available to help us update our state estimate between time k and time $k+1$, so we should just update the state estimate based on our knowledge of the system dynamics [32].

Obtaining the state estimate for the next time iteration solely is not enough for the following data step. As noted before, the covariance matrices play an important role in the performance of the KF. That is why the propagation of a state covariance matrix, $P(k+1|k)$, is needed too.
In this case an important stability equation has to be introduced. If the state matrix A is stable, the matrix P converges to the steady state covariance, which satisfies the Lyapunov equation [13]

$$P(k+1) = A(k)P(k)A(k)^T + Q(k) \quad , \tag{4.10}$$

where in the general definition matrices P, Q are symmetric and for any Q, which is positive definite, there exists a unique matrix P, which is also positive definite. In the case of KF, we are using the state and noise covariance matrices.
Equation 4.10 is a discrete Lyapunov equation. For the continuous time application, the equation has a slightly different form:

$$\dot{P}(t) = A(t)P(t) + P(t)A^T(t) + Q(t) \quad , \quad P(t_0) = P_0 \tag{4.11}$$

In our case, the used model has been continuous, we have to use equation 4.11 for the state covariance matrix propagation:

$$P(k+1|k) = P(k|k) + (AP(k) + PA^T + Q)dt \quad . \tag{4.12}$$

4.5 Extended Kalman filter

The EKF is a needed tool to use when working with non-linear systems. Given the continuous nature of dynamical systems and the requirement of microprocessors for discrete data, the continuous–discrete EKF is usually the most useful formulation for modern control purposes [21]. The main idea is a linearisation of a function in every step to provide estimates in small steps, which allows us to deal with some non-linearities. Linearisation is a very common engineering way of constructing approximations to non-linear systems and thus it is very easy to understand and apply. The advantage of the EKF over other non-linear filtering methods is its relative simplicity compared to its performance. However, with its advantages there are some disadvantages, too.
A disadvantage is that because it is based on a local linear approximation, it will not work in problems with considerable non-linearities. The filtering model is also restricted in the sense that only Gaussian noise processes are allowed and thus the model cannot contain, for example, discrete valued random variables [33].

The requirements for the EKF we are using in this work, on top of the KF assumptions listed earlier are [33]:

- The measurement model and its dynamic model functions are differentiable
- The process and measurement noise is assumed to be additive

In comparison with the KF, the EKF does not use the original system matrices $A - D$. The EKF basic equations are:

$$\begin{aligned} x_{k+1} &= f(x,u) + v(k) \\ y_k &= g(x,u) + e(k) \end{aligned} \quad . \tag{4.13}$$

where $f(x,u)$ is a non-linear state equation including the input relation, $g(x,u)$ is an output equation containing both states and inputs and v, e are the state, resp. measurement noises.

Due to the non-linearities, in order to get the state matrices, we need to compute a Jacobian of the functions f, h in each time iteration. Also, after the Jacobian is obtained, we substitute the best available state estimate we have, $\hat{x}$, from the data in previous time step, for x and the deterministic input value u for the state x and the input u. Following the basic KF algorithm, the noises are assumed additive, thus are added to the resulting equations. To determine a first-order approximation. Representing the Jacobian as The resulting matrices are obtained as follows:

$$\begin{aligned} A &= \left.\frac{\partial f(x,u)}{\partial x}\right|_{x=\hat{x}(k|k-1);u=u(k)} \\ B &= \left.\frac{\partial f(x,u)}{\partial u}\right|_{x=\hat{x}(k|k-1);u=u(k)} \\ C &= \left.\frac{\partial h(x,u)}{\partial x}\right|_{x=\hat{x}(k|k-1);u=u(k)} \\ D &= \left.\frac{\partial h(x,u)}{\partial u}\right|_{x=\hat{x}(k|k-1);u=u(k)} \end{aligned} \tag{4.14}$$

The matrices C, D are obtained during the data-update step and matrices A, B are the Jacobians from the time-step. After getting the Jacobians, we continue the algorithm as described in the chapter Kalman filter algorithm. Had the noises assumed not been additive, we would have to define the noise matrices $\Gamma_{v/e}(k)$, through the linearisation process and modify (extend) the eqs. 4.8 and 4.12.

4.6 Goals of the Kalman filtering

There are two main goals of the Kalman filtering we want to achieve:

- Estimate the side-slip angle of the vehicle β
- Estimate the constant DF_z in the Pacejka's Magic formula

The estimation of the side-slip angle is to be done using only the measurements of vehicle's forward velocity v and the yaw rate $\dot{\psi}$ and the known input of wheel steering angle δ, driving torque τ and the braking torque τ_B. There is an option to refine the measurement with the wheel's rotational velocities $\dot{\rho}$ to gain an additional information about the vehicle for more precise estimation, but it is not necessary, since the knowledge of v and $\dot{\psi}$ from the state vector is sufficient. In comparison with the latter goal, in this case we know the wheel load F_z and the Peak factor in the PMF, D.

The second task is more general, with the reference to the state vector. The main goal is to estimate a part of the PMF, which describes the forces acting on tyres both longitudinally and laterally. Considering the wheel load F_z is often set as a constant, it does not hold the same value on any surface, which might result into incorrect estimations of tyres' behaviour. That is why a constant DF_z was chosen, in order to be able to estimate a sudden change in the tyre behaviour (as an example might be drifting from asphalt road to sand/grass). Having the adaptive estimation is very beneficial, because the wheel slip angle α describing the tyre dynamics is conditioned by a vehicle side-slip angle β - with the β estimated correctly for the full duration of the ride, we have an accurate information about the vehicle travelling on various surfaces.
For this task we need more measurement data, that is why the wheel rotational velocities $\dot{\rho}$ are added to the measured output.

Chapter 5

Experiments

This chapter provides the insight into the application of previously mentioned main points - system modelling and Kalman filtering. Some of the experiments are designed to demonstrate the filtering/smoothing function of KF and the comparison of the performance between the non-linear and the linearised models. The main part focuses on the estimation of vehicle's states under different conditions and manoeuvres. It is divided into two main parts:

- Virtual experiments (Simulation results)
- Validation on real measured data

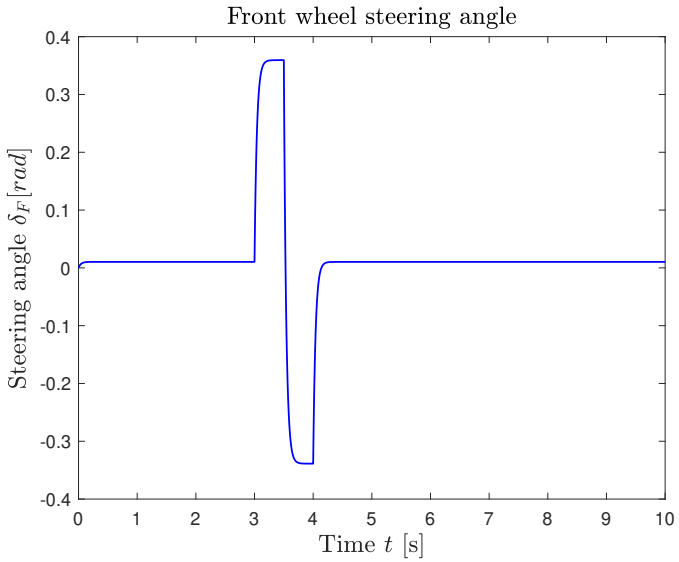
5.1 Simulation results

5.1.1 Design of the experiments

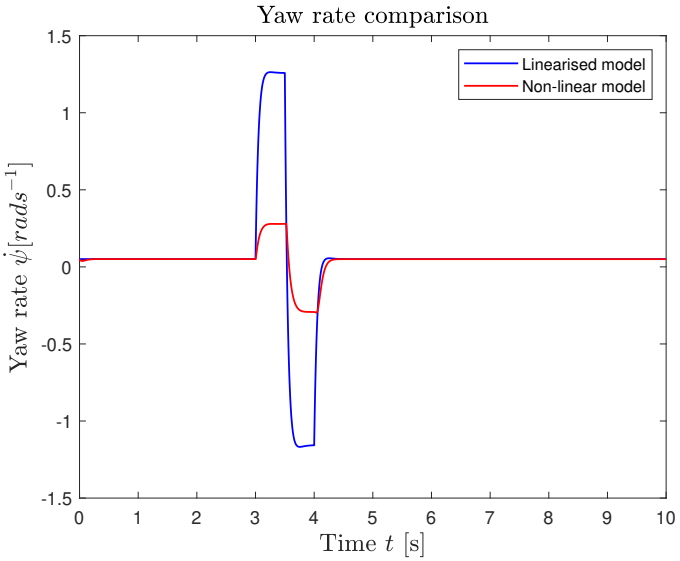
There have been two main parts in the simulation experiments. First part was a comparison between a linearised and non-linear model. All simulations were done in Matlab, where scripts for signal processing and KF algorithm were implemented. The models from Ch.2: System modelling have been used in Simulink.
Secondly, we compared the KF and EKF used as a filtering element in a very noisy environment (artificially added white noise with large SNR). As an input we chose a doublet (A crucial aspect in KF-tuning is the choice of manoeuvres/input signals. From a more practical point of view, a "wide excitation spectrum" shall be used [8]) on the steering angle to show the model reactions, the car configuration selected was of Skoda Fabia. The input signals differed during the model comparisons, starting from the doublet, followed by the same-amplitude sine wave, all the way up to an increasing amplitude of the sine wave. Only one of the mentioned inputs is presented as it is sufficient for the demonstration purposes.

5.1.2 Linear and non-linear model comparison

As noted in Ch.3: Data generation, we had a non-linear model and also its linearised (simplified) version for comparison. The linearised model describing the tyre behaviour consists of two parts, two saturations and a linear function connecting them, whereas the non-linear is defined by the Pacejka's magic formula. The results of this small difference on the scale of tyres all the way up to the full vehicle states is seen in this comparison, which also highlights the need for a non-linear model use.



(a) : Front wheel steering angle test input



(b) : Vehicle yaw rate comparison

Figure 5.1: Non-linear and linearised model comparison

As we can see from fig. 5.1, the linearised model reacts much slower, than the non-linear one. Also, about a three-times larger yaw rate has been modelled by the linearised model, the non-linear holds the steering angle in the amplitude too. These experiments have been produced from the setup of a steady straight ride with a vehicle velocity $v = 20ms^{-1}$, the front wheel steering angle was $\tau_F = \pm 20° \approx 0.35\ rad$ over one second.

The linearised model does not match well, because during the linearisation we assumed low steering angles (up to 10°). The larger steering angle was set on purpose, because the measured data, where we are using the EKF, have the steering angles around 25°. The linearised model has been set out of the linearisation area. This also highlights the constraints, that are imposed on the model using simplifications in order to get rid of the non-linear functions - for accurate estimation in our case it is not sufficient using the linear KF.

5.1.3 Filtering and smoothing

When set correctly, the KF can work really well even with very noisy signals. There are big differences between the measurement noise and the process noise. Large values of process noise can change the system in a way, where the initial modelling approach fails. Then no KF would work. In real world, the noisy measurement is much more common. Therefore we have designed an experiment with added white noise on measurements, where all we got was data with high SNR (in this case the SNR of measurement white noise was set to 30 dB), the initial conditions and the inputs. The goal was to filter the noisy values and compare it to the original model output (without added noise). Also, an estimate of the vehicle side-slip angle β was done in order to show how the estimate was affected by high noise levels.
Figure 5.2 shows several observations. Firstly, the noise totally destroyed the measured value, and if it were the only measurement, KF would not be able to make good results solely out of this noise. Secondly, as we can see, in this case the KF output relies heavily on the created model (we could see from the earlier comparison that the linear model is not as precise as the non-linear). This reliance is needed since the measurements are in this case useless: measurement covariance noise matrix R is set to a high value, denoting high measurement noise value, and the process noise covariance Q is set as very low in order to rely on the model. Lastly, even though the EKF results look fine, they are still deformed in shape and offset from the noise-free signal. This is also the result of the large noise values, causing the filter results to be affected from the noise covariance matrices setup.

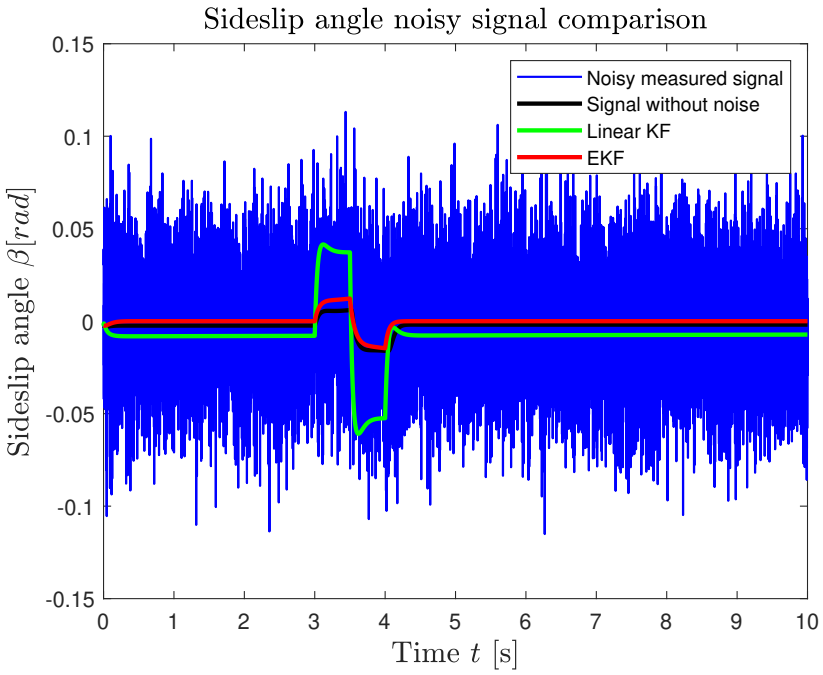


Figure 5.2: Filter performance in a very noisy signal

5.2 Results validation on the real data

For the real data validation we used the data measured on the FEE eForce formula. The trajectory was several turns forming a number eight in order to get larger steering angles with the steering being done in the same way both to the left and right side. The vehicle velocity differed throughout the experiment, which was also useful, since we could just pick a part where the steering angles are large and velocity is not too low.

The full data obtained from the on-board GPS, are measured up to the time $t = 800s$, but in our case, we are using the data in the middle of the experiment (time roughly between $t = 400 - 500s$, in green/yellow), because the velocity of the vehicle was in compliance with our assumptions mentioned earlier ($v > 5ms^{-1}$), and the steering performed was fluent, periodic and in full range of the measured data. All inputs from the driver are shown in detail in the next section.

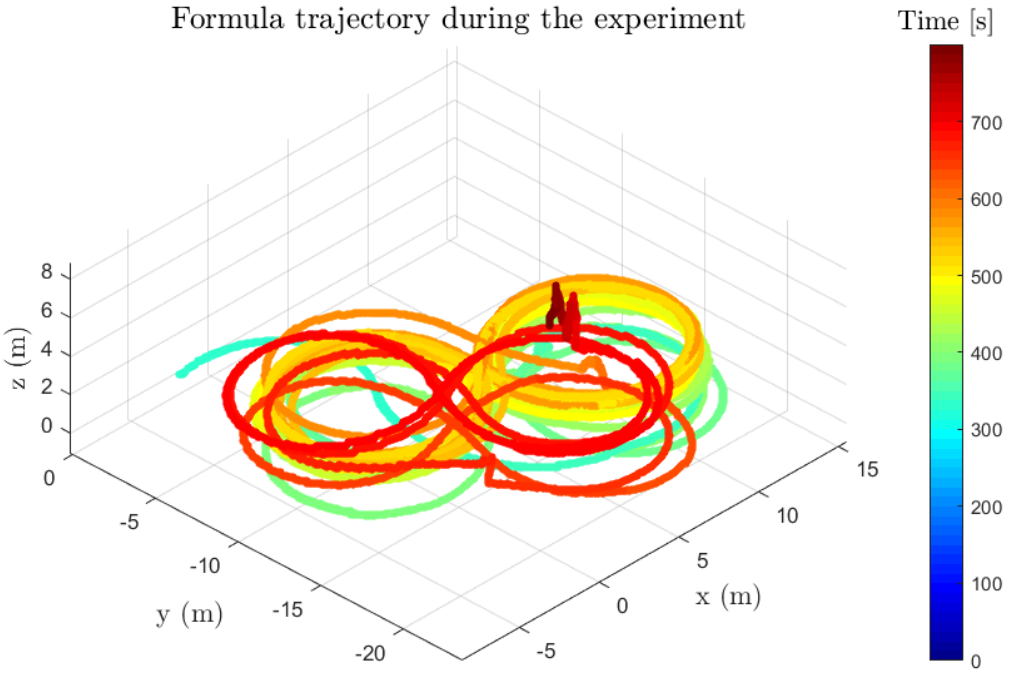
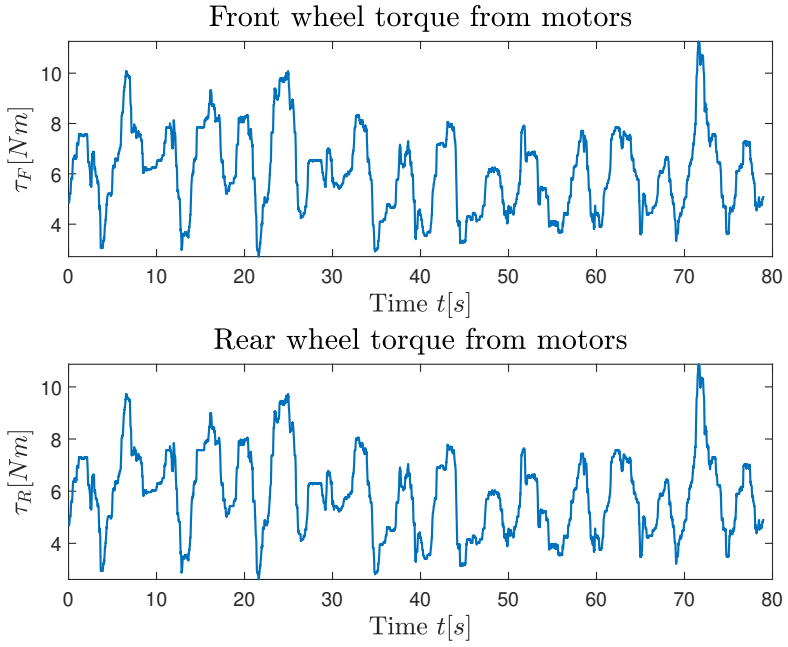


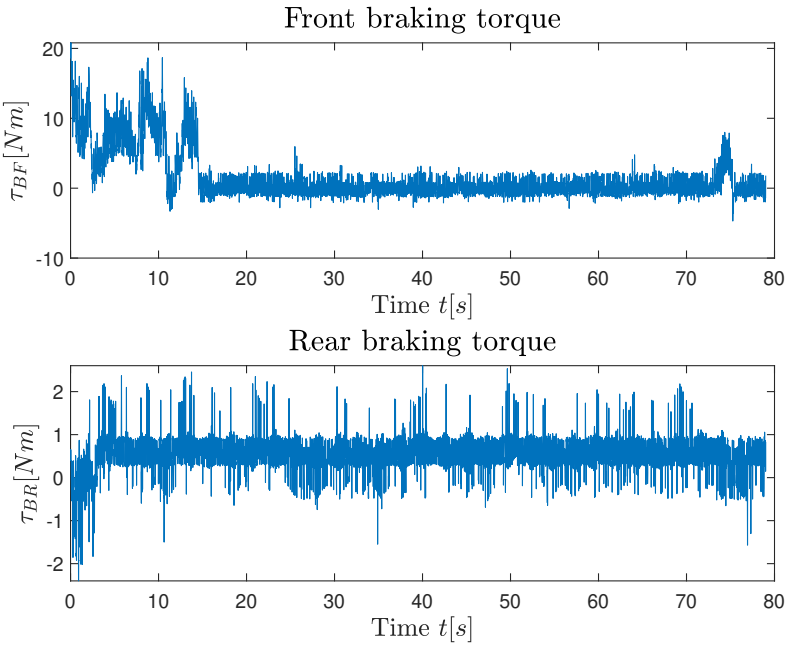
Figure 5.3: The formula elapsed trajectory

5.2.1 System inputs

There were three sets of inputs from the driver. The front wheel steering angle δ_F, which was in the range of $-25°$ to $+25°$, and because of the model contains also rear steering angle, which was not used, it was set to zero and is not shown in the figures.
The driving torque for front and rear wheels $\tau_{F,R}$, which are almost the same, these values are a mean value of the two motors (one for each wheel) on the front/rear side of the vehicle, because of the S-T model use.
The braking torque applied from brakes $\tau_{BF,R}$ was engaged at the beginning of the chosen range mostly on the front brakes. The rear brakes were not used in this part and the shown data contain mostly noise. Torque inputs can be seen in fig. 5.4 and wheel steering angle in fig.5.5.



(a) : Front and rear wheel driving torque



(b) : Front and rear wheel braking torque

Figure 5.4: Formula driving and braking torques comparison

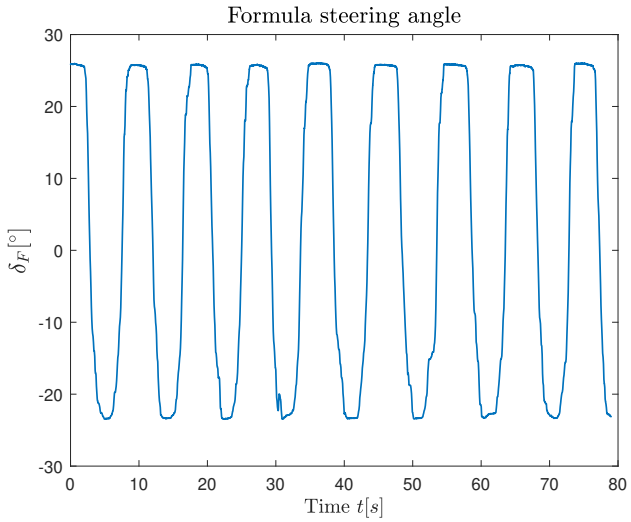


Figure 5.5: Formula front wheel steering angle

5.2.2 Model fitting

Model mismatch can create a total KF collapse and therefore it is imperative to check if the created model produces similar outputs when fed the same input signals. For comparison, we have picked the values in a measurement vector, the vehicle velocity v, its yaw rate $\dot{\psi}$ and the sum of front and rear wheel rotational velocities $\Sigma\dot{\rho}_i$.
Even if the beginning of the comparison shows a difference, mainly in the vehicle velocity and the wheel rotational velocities, the most of the modelled shape fit quite well. The difference at the beginning is due to a specific chosen time window, where previous development of data is not shown. The yaw rate is matched very well.

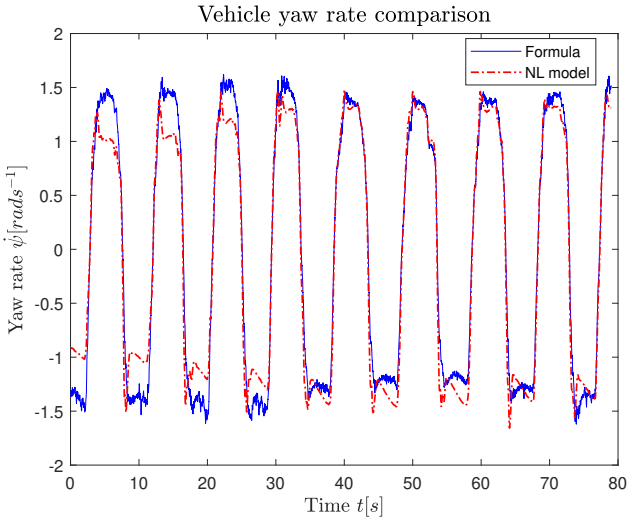
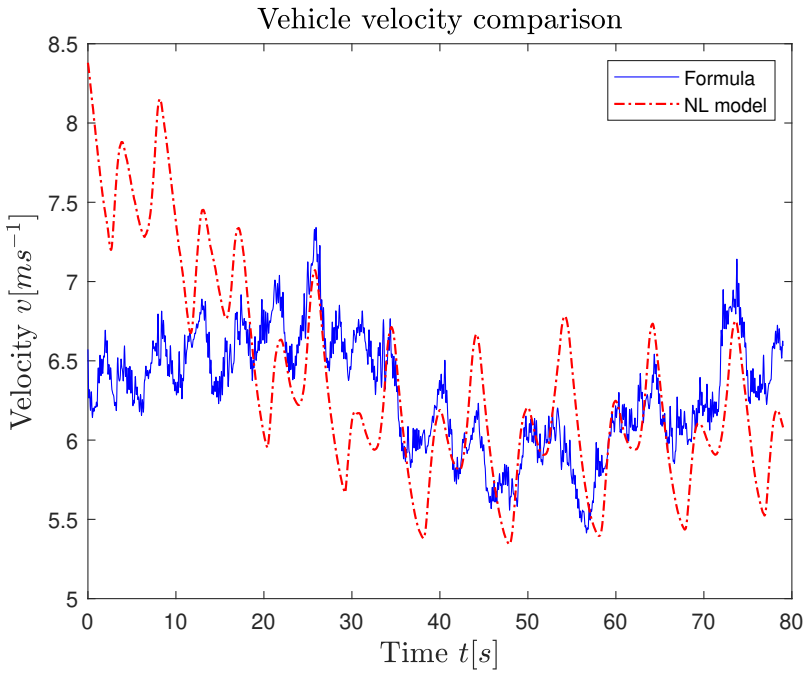
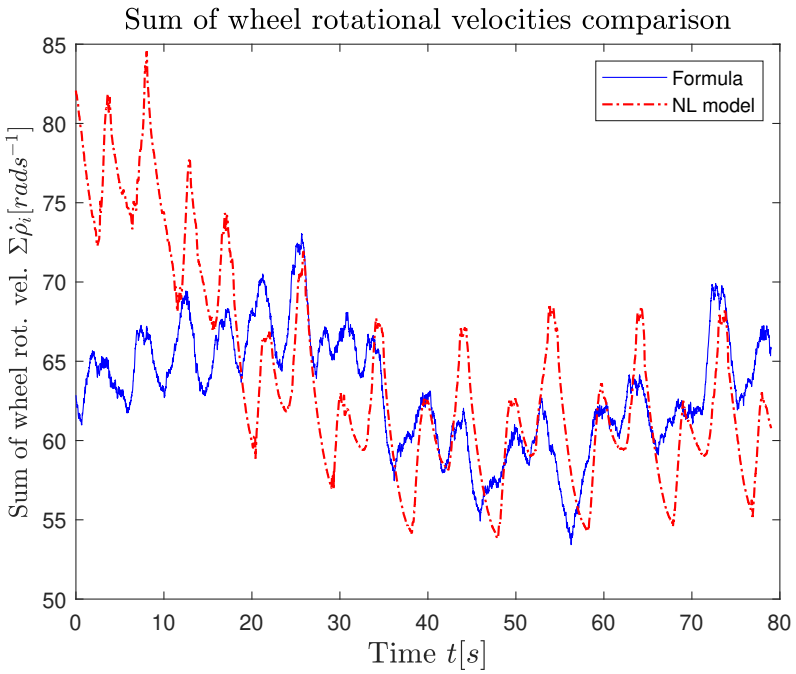


Figure 5.6: Yaw rate non-linear model comparison with formula measurements



(a) : Vehicle velocity comparison

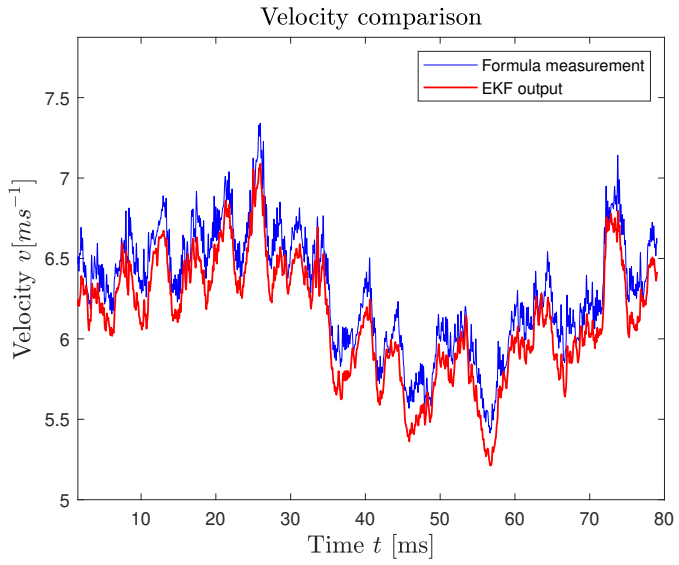


(b) : Sum of wheel velocities comparison

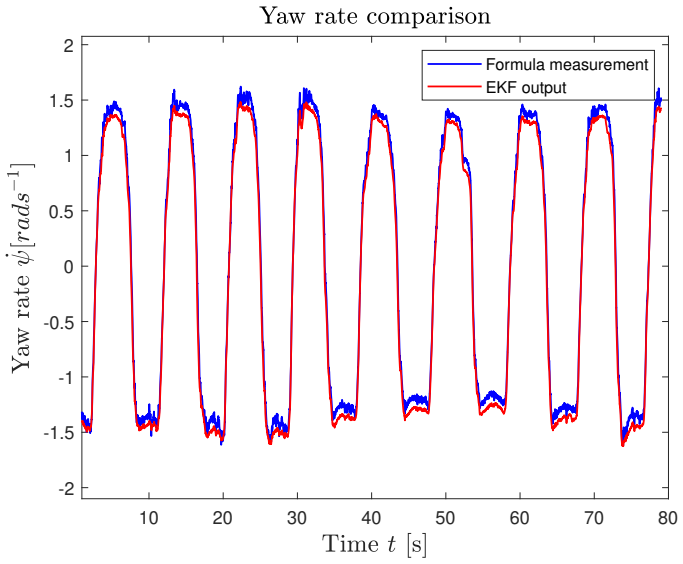
Figure 5.7: Non-linear model comparison with formula measurements

5.2.3 Filtered states

This part is the comparison between the output values of the EKF and true measured data on the formula. Using specific values of noise covariance matrices Q, R allows us to move the filter performance either towards the smoother signal or more accurate (almost copying measured values). The aim was to find a nice approximation of the noisy measured signal without following the additional noise. Reducing the noise enables us to use the signal for the purely approximated states and values, which are defined by them. As we can see from fig.5.8, the noisier of the signals was the vehicle velocity.



(a) : Velocity of the vehicle from EKF



(b) : Vehicle yaw rate from EKF

Figure 5.8: EKF measurable states comparison

Apart from the states shown, we also had the wheel rotational velocity $\dot{\rho}$ as an additional state. Its trend was almost the same as the CoG velocity (can be seen in fig. 5.7), therefore only one of the plots is shown. Velocity was chosen because the measured value was with more added noise, thus making the smoothing more visible.

5.2.4 Estimated states

The estimated states from the EKF run were:

- Vehicle side-slip angle β
- Front wheel peak factor and wheel load combined $D_x F_{zf}$
- Rear wheel peak factor and wheel load combined DF_{zr}

The Kalman gain with covariance matrix setup plays an important role in states estimation. It is not easy to correctly set up the noise covariance matrices the more states are present. Some helpful observations from statistical point of view are described in the next chapter, Ch.6: Performance evaluation.

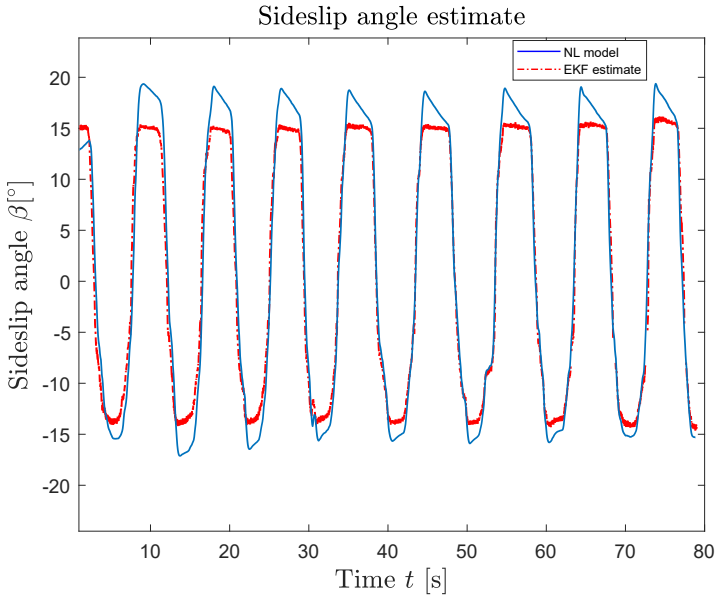


Figure 5.9: Estimate of vehicle side-slip angle

From fig. 5.9 we can see, that the vehicle side-slip angle β follows the wheel steering angle in shape, also, the measured vehicle yaw rate is in comparison slightly shifted to the left - this is the delay, before the steering of the whole vehicle happens. This is the first purely estimated state and we can say that the estimation is pretty accurate, for the reasons mentioned earlier and in comparison with the model internal value of β. It is not possible to compare this estimate plot directly with some measured data from the formula, because the vehicle side-slip data was not measured.

Next estimation goal was to estimate the values of peak factor of the front, D_x, resp. rear wheel (D) combined with the vehicle wheel load $F_{zf,r}$. This is used as a constant value in the Pacejka's tyre model, eq.2.13. Being able to estimate this value we can react to changes on the road surface (the μ differs - there is a large difference between icy road and the dry asphalt). These changes can make the modelling inaccurate if using constants measured during specific conditions. Moreover, the tyre peak factor D is different on many tyres and the true value might not be available - sometimes the Pacejka's constants, which characterise the tyre, are secret of the tyre manufacturer.

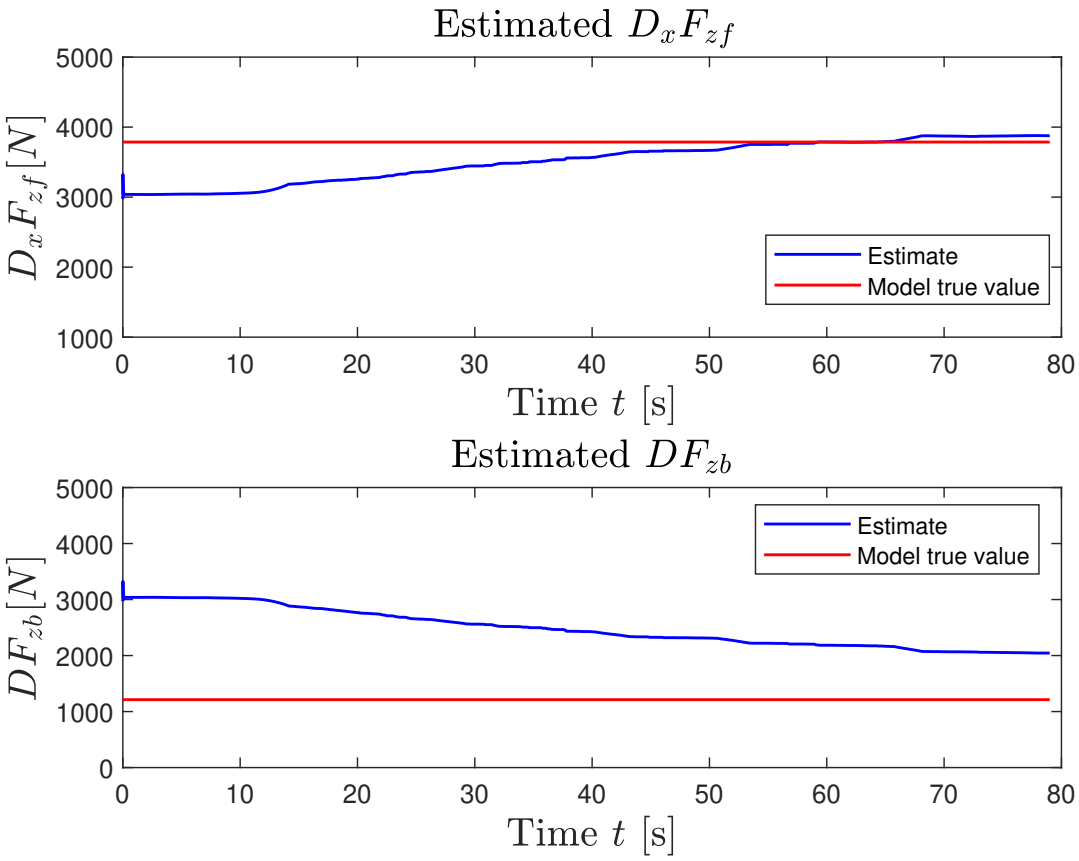


Figure 5.10: Estimate of the peak factor and wheel load

From the initial conditions set by guess as $DF_z = 3000N$ we can see the values of peak factors used D and the wheel load for each wheel, F_z, which are compared to the estimated value of this combined "constant". The estimate converges over the time to a value, which is used in the tyre forces Pacejka equations. The real wheel load is not that different in value, but what makes the combination DF_z distinct for each wheel is the value of peak factor D. This is why the front wheel (first subplot) has a higher value to converge to than the initial conditions, the peak factor D_x was used in the model as 2.5.

The other estimating part are values, which are not directly estimated, but computed from the estimate of tyre forces, namely the tyre slip angles α. The estimate shows very low tyre slip on the front tyre, this is because the tyre is being steered itself by the input tyre steering angle. Where relatively speaking larger slip occurs, is the rear tyre, because the formula has steering on front wheels only. The slip angle is ideally zero, meaning the car holds the set steer, this value would be interesting for control systems in the possible work.

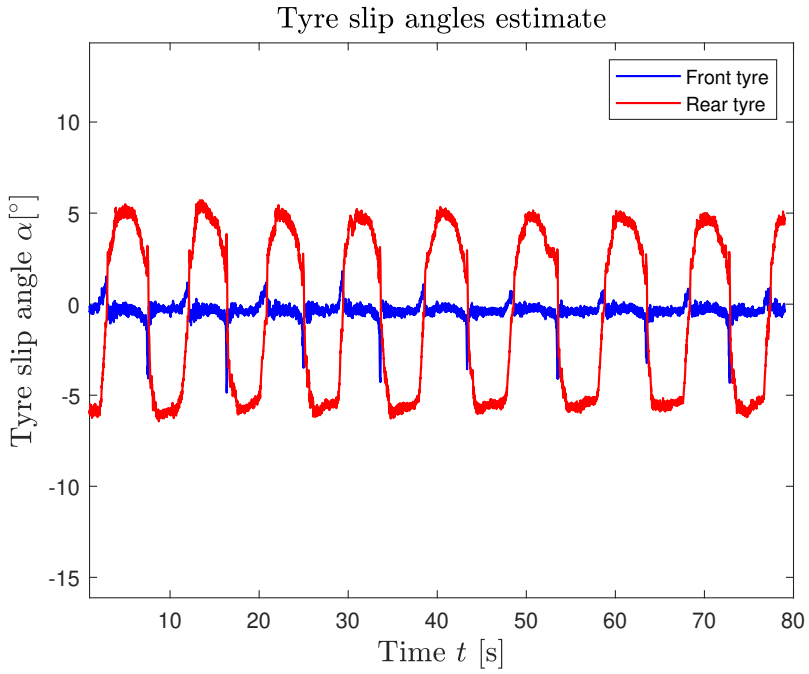


Figure 5.11: Estimate of tyre slip angles

The problem with accuracy was encountered during the tyre slip ratio λ. Due to the use of S-T model, all wheel variables have been averaged from the two tyres' data. During the turning manoeuvre the front wheels behave in the opposite way (one is accelerating and the other is decelerating). This results into positive slip ratio on one wheel and negative on the other. Averaging both wheels in that case does not achieve the simplification of real behaviour, but creates some false information instead resulting into a small model mismatch. The problem has been dealt with manually by fitting the model to behave appropriately, the solution is suggested in the future work: the implementation of twin track model.

Chapter 6

Performance evaluation

Kalman filter performance can be seen directly in the resulting plots in a comparison with original, measured values. Quantitatively, this comparison of trends can be expressed via Goodness of fit, which is using the *innovation* to express how much does the KF output differ from the measured value. Sometimes, it might be useful to look at the performance from a different point of view - testing that the assumptions made still stand and the basic, necessary rules defining the estimate:

- Whiteness (stochastic properties are unaffected with a deterministic input)
- Consistency (estimate improves with a sample size)
- Unbiasedness (the expected value is the same as the true value of parameters)
- Efficiency (best of all available estimates, Cramer-Rao bound)

There are several tests, which are using statistical information and can verify these properties, which are very well described in [3] and [18]. These tests can use the data from various Monte Carlo simulation runs, but are also applicable for a single-run (which we are using here). The tests can help during the tuning of the KF noise covariance matrix coefficients too. The decision making for the tuning (the information value of the test results) is described later on for each test. Naturally, not every time have the initial conditions been set up using all of these tests together, the tests serve more as a feedback tool rather than a strict-tuning rule.

Unfortunately, in order to perform these tests, we need some measured data to compare it with to get the innovation. This is not possible in the case of estimated states (β, DF_z), that is why the tests are performed on measured states and in the event of success we can say, that the estimates should be performing well.

6.1 Data-driven and whiteness test

In this part the results of EKF estimates are compared to the measured values. The metric we are using is normalized root-mean-square error (NRMSE), which is defined as follows:

$$NRMSE = 1 - \frac{\sqrt{y^2 - \hat{y}^2}}{\sqrt{y^2 - \bar{y}^2}} \tag{6.1}$$

The data-driven test is a simple check of how well is the estimator able to copy measured values. The fit is shown in percentages NRMSE. In a case of high noise it might be better to have smoother fit through the noisy data, therefore a value of 100% is not always needed for a good filter performance, which can be evaluated more in-detail using the some of the tests described later.

Stochastic properties of the KF should be not affected by deterministic inputs. If we assume the added noise to be white with a zero mean, we want to check back after the KF run if this property has not been affected by the inputs entering the system. To test this, a very common test exists for the whiteness of the innovation $\nu(t)$ check. It uses a sample autocovariance function $R_{\epsilon\epsilon}(t)$. The autocovariance is usually normalised by the largest value $R_{\epsilon\epsilon}(\tau)$. To pass the test we expect a peak around $\tau = N/2$, which will be located in the middle of sample range, since the autocovariance is symmetric around the axis y, and we shall see random distribution around zero. For large sample size we assume that $R_{\epsilon\epsilon}(t) \in \mathcal{N}(0, 1/N)$. From that the 2σ boundary can be said to be $\pm 2\sigma = \pm 2/N$, and we check again if at least 95% of the autocovariance values fall within the region [18]. The result of this test can also be used as a part of consistency check.

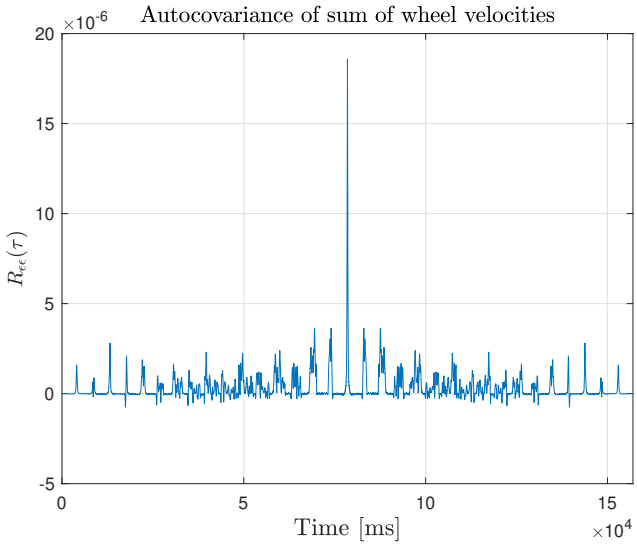


Figure 6.1: Sum of wheel velocities whiteness test

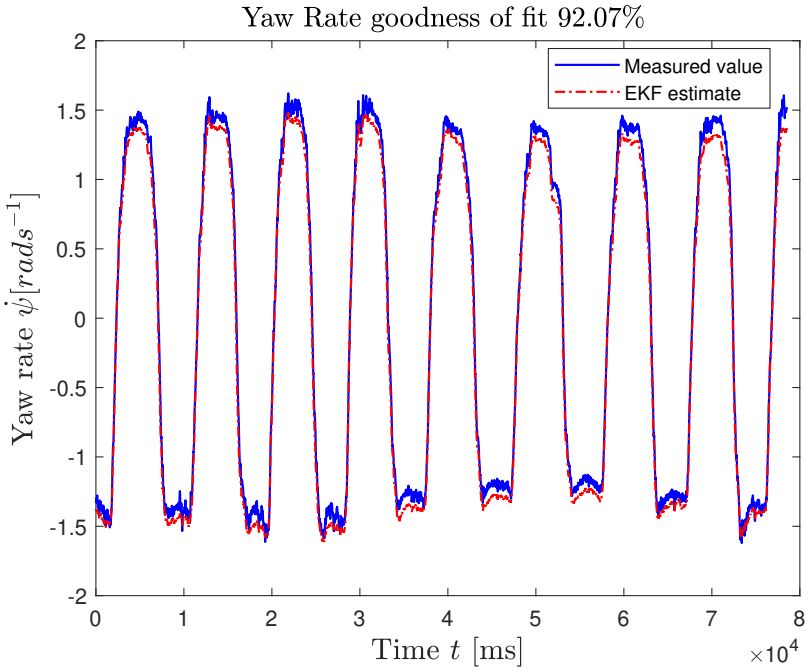


Figure 6.2: Yaw rate goodness of fit

6.2 Consistency testing

Consistency of the estimate reflects how well the estimated probability distribution of the parameters agrees with its true distribution [19]. Since the filter gain is based on the filter-calculated error covariances, it follows that consistency is necessary for filter optimality: Wrong covariances yield wrong gain [3].
This filter property can be tested using the innovation and normalised innovation squared (NIS). The NIS uses the difference of measured output and estimated output, the innovation ν and the state covariance matrix element P_{yy}, which is also a covariance matrix of the innovation. Filter can be passed as consistent, when the following criteria, which can be tested in applications with real data, are applicable [3]:

- The innovations should be acceptable as zero mean and have magnitude corresponding with the state covariance as yielded by the filter.
- The innovations should be acceptable as white.

The whiteness test was shown earlier, and other test for consistency can be the innovation magnitude bound (IMB) test. This test uses the basic innovation and simply checks, how do its values differ from the mean. The mean, as assumed, should be zero and we expect about 95% of the innovation magnitude to fall within the $\pm 2\sigma$ bounds. If it stays in, we can say the filter estimate is consistent.

If the IMB test results are well within the defined $\pm 2\sigma$ bounds, then the noise matrices Q, R are guessed high. Same applies for the opposite, too low Q, R result into exceeding the bounds. As for the autocovariance test, this can decide between Q and R, what needs to be tuned. Additionally, if the autocovariance shows no time correlations (looking like Dirac in the middle and the rest is distributed randomly around zero), the R is low or Q is high. Fig. 6.3 shows the IMB test performed on the velocity of the vehicle.

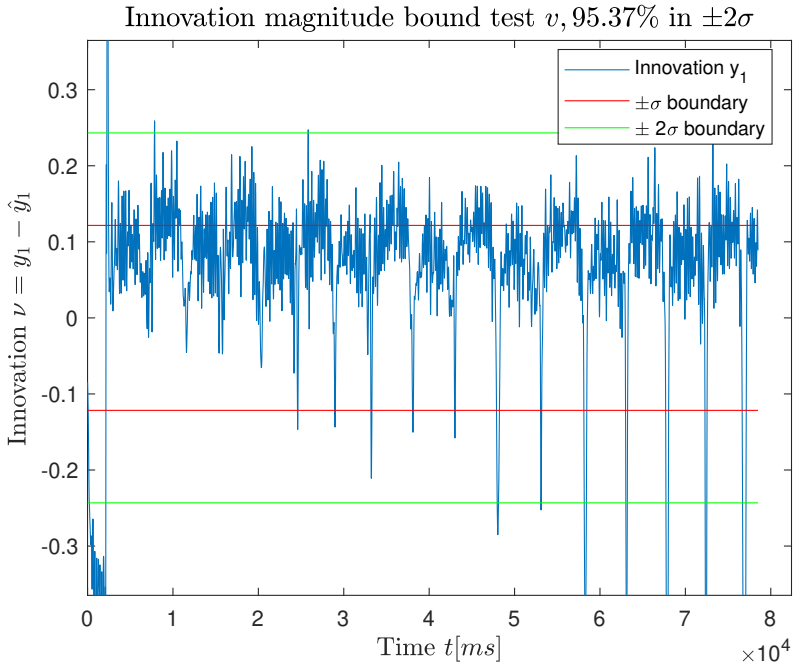


Figure 6.3: Innovation magnitude bound test of the vehicle velocity

6.3 Unbiasedness testing

The innovation magnitude bound test is checking the unbiasedness of the innovations too. If approximately 95% of the values are within $\pm 2\sigma$ range, we can say the innovation is unbiased. This test is checking for filter consistency too, however, it is better used with the Normalised innovations squared test (χ^2) in combined as that test is more strict in terms of unbiasedness testing.

The chi-square (χ^2) NIS test uses the normalised form of innovation squared. The test is concluded from hypothesis H_0 that a mean innovation squared ($\bar{q}$) multiplied by the number of samples (N), $N\bar{q}$, has the distribution of χ^2_{Nm} with a probability of: $(1-\alpha)$. In our test we picked $\alpha = 5\%$. We also need to find borders $[r_1, r_2]$ fulfilling [18]:

$$\begin{aligned} P(N\bar{q} \in [r_1, r_2] | H_0) &= 1 - \alpha \\ [r_1, r_2] &= \left[\chi^2_{Nm}\left(\frac{\alpha}{2}\right), \chi^2_{Nm}\left(1 - \frac{\alpha}{2}\right) \right] \end{aligned} \tag{6.2}$$

If the result of χ^2 NIS test is outside the defined interval, then there is some space for further KF tuning. If the total $N\bar{q}$ is below the $[r_1, r_2]$, that means that the noise covariance matrices Q, R are guessed too high. Similarly, if the total $N\bar{q}$ is higher than $[r_1, r_2]$, the matrices Q, R are guessed too low. Then we can focus on particular states' IMB and whiteness tests to determine how does the relevant noise matrix diagonal element stand.

6.4 Efficiency testing

Efficiency can be tested using methods in [3]. The main point is to prove the estimate is Cramer-Rao bound using the Fischer information matrix values for the test. For practical reasons, the tests described earlier, the consistency, unbiasedness and whiteness tests proved to be useful also in the process of tuning and not only evaluation of the KF. That is why they are explained more in detail to show the connection between the noise covariance matrices and the statistical properties of the innovation.

Chapter 7
Conclusions

The thesis had two main parts, estimation of vehicle states (including the side-slip angle, which was not originally measured) and tyre states. These have been chosen as the tyre slip angles and the wheel load with peak factor combined. All of the goals above have been successfully accomplished using the Extended Kalman filter and the created non-linear vehicle Single-track model. The estimate of DF_z shows a very accurate estimate, if the initial condition is not set far from the real value, which can be seen on the front wheel results. The rear wheel had an initial condition more than twice larger than the real value, the convergence can be seen, but is not fast enough to reach the real value, desired convergence time can be thought as being $10\times$ faster than the dominant pole of the vehicle. An accuracy issue was encountered during the slip ratio estimation, because of the wheel averaging needed for the used Single-track model. The solution to this issue is addressed in the next section, the more complex Twin-track vehicle model is needed.

7.1 Summary

In *Ch.2: System modelling* we have demonstrated the principles of Single-track vehicle model, modelling both the vehicle centre of gravity behaviour and also the tyres' properties. *Ch.3: Data generation* described the models used for data generation, both non-linear and the linearised, which serves for comparison and the introduction of electric formula, which was used for model fit. Necessary conversions of the measured data have been made in order to match the model coordinate systems and units. *Ch.4: Kalman filtering* explained how the Kalman filter task works in its basic principles, its abilities and introduced the Extended KF for non-linear systems approximation. *Ch.5: Experiments* presented the achieved results, both from simulation and also in comparison with real, measured data. The main states, which have been set up as goal (the vehicle side-slip angle, the tyre slip angle and the combination of peak factor with the wheel load), have all been successfully estimated. Finally, *Ch.6: Performance evaluation* introduced several methods of the basic estimate properties check and explained the process of the noise covariance matrices initial setup, its effects on the KF results and provided a statistical view on the achieved results.

7.2 Future work

- **Implementation of the Twin-track model**: Using the Single-track model can lead to simplifications, which would affect the accuracy of modelling. Starting with the tyre steering angles, these are not the same for both wheels. As an example are the slip ratios λ. Averaging of wheels oversimplifies the tyre slips, resulting into less accurate estimation. Another case can be seen during cornering: the "inner" wheel has larger steering angle by about 15% compared to the "outer" wheel.
- **Comparison of the performance with unscented KF**: The EKF takes only one point from the Gaussian distribution (mean) and approximates the non-linear function via linearisation. The unscented KF is based on several points, the sigma points, which are then approximated. This could lead to more precise approximation and thus a comparison of EKF performance with unscented KF comes forward as an extension possibility of this thesis.
- **Estimation of noise covariances**: In linear stochastic state-space representations, the true noise covariances are generally unknown in practical applications. Using estimated covariances a Kalman filter can be tuned in order to increase the accuracy of the state estimates [1]. So far, the matrices has been "guessed" after a lengthy tuning and remained constant.
- **Computational demands analysis**: The use of more accurate methods might be very demanding, as far as the computational time is concerned. When used in embedded applications, faster algorithms might be preferred to the ones, which are more accurate, but slower. The comparison of these factors could be a part of feasibility study, deciding which method will be used.
- **Design of the vehicle control system during skidding**: Using the estimates of tyre slips can be useful to build on during the design of vehicle safety control system, preventing drifts of the vehicle by using the optimal control, based on measured and estimated data.

Appendix A

Bibliography

[1] B.M. Åkesson, J.B. Jørgensen, N.K. Poulsen, S.B. Jørgensen, *A generalized autocovariance least-squares method for kalman filter tuning*, Journal of Process Control **18** (2008), no. 7, p. 769–779.

[2] J. Ackermann, *Robust Control: Systems with Uncertain Physical Parameters*, 1st ed., Springer, London, 1993.

[3] Y. Bar-Shalom; X. Rong Li; T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley and Sons, New York, 2001.

[4] G. Cai; T. H. Lee; B. M. Chen, *Unmanned rotorcraft systems*, 1st ed., Springer-Verlag London, 2011.

[5] V. Cossalter, *Motorcycle dynamics*, Lulu.com, 2006.

[6] D.M.Bevly; J.C.Gerdes; C. Wilson; G. Zhang, *The use of GPS based velocity measurements for improved vehicle state estimation*, Proceedings of American Control Conference (ACC), 2000.

[7] M. Doumiati; A. Charara; A. Victorino; D. Lechner, *Vehicle Dynamics Estimation using Kalman Filtering: Experimental validation*, Wiley-ISTE, 2012.

[8] J. Edelmann; M. Plöchl; M. Haudum; M. Höll, *Vehicle state estimation from a sports-car application point of view focusing on handling dynamics*, Advanced Vehicle Control (2016), p. 521–526.

[9] eForce FEE Prague Formula, `https://eforce.cvut.cz/`.

[10] D. Efremov, *Unstable ground vehicles and artificial stability systems*, Czech Technical University in Prague, 2018.

[11] J. Filip, *Trajectory Tracking for Autonomous Vehicles*, Czech Technical University in Prague, 2018.

[12] Y. Fukada, *Estimation of vehicle sideslip with combination method of model observer and direct integration*, Proceedings of International symposium of Advanced vehicle control (AVEC), 9 1998.

[13] Z. Gajić and M. T. J. Qureshi, *Lyapunov Matrix Equation in System Stability and Control*, Academic Press, 1995.

[14] L. Haffner, *Real-time tire models for lateral vehicle state estimation*, from Universitätsbibliothek der TU Wien, 2008.

[15] V. Havlena; J. Štěcha, *Moderní teorie řízení*, Ediční středisko ČVUT, 1999.

[16] M. Hiller, *Eine Einführung in die analytische Mechanik und Systemdynamik*, Springer-Verlag Berlin Heidelberg, 1983.

[17] Y. J. Hsu; S. M. Laws; J. C. Gerdes, *Estimation of tire slip angle and friction limits using steering torque*, IEEE transactions on control systems technology **vol.18** (2010), no. 4, p. 896–906.

[18] I.Reid, *Lecture notes for performance check, Estimation II*, Hilary Term, 2001.

[19] P. Ivanov; S. Ali-Löytty; R. Piché, *Evaluating the consistency of estimation*, Proceedings of 2014 International Conference on Localization and GNSS, 06 2014, pp. 1–5.

[20] U. Kiencke and L. Nielsen, *Automotive Control Systems: For Engine, Driveline and Vehicle*, 1st ed., Springer-Verlag, 2000.

[21] F. L. Lewis; L. Xie; D. Popa, *Optimal and robust estimation: with an introduction to stochastic control theory*, 2nd ed., CRC Press, 2007.

[22] M. Lászlo, *Flight Control Solutions Applied for Improving Vehicle Dynamics*, Czech Technical University in Prague, 2019.

[23] Skoda Fabia Owner's manual, `http://ws.skoda-auto.com/OwnersManualService/Data/en/Fabia_54/05-2011/Manual/Fabia/A05_Fabia_OwnersManual.pdf`.

[24] L. Meirovitch, *Dynamics and Control of Structures*, John Wiley and Sons, New York, 1990.

[25] D. L. Milliken; E. M. Kasprzak; W. F. Milliken; L. D. Metz, *Race car vehicle dynamics: Problems, answers and experiments*, Premiere Series Books, 2015.

[26] M. Mitschke and H. Wallentowitz, *Dynamik der Kraftfahrzeuge*, 4. ed., Springer-Verlag Berlin, 2004.

[27] UK Oxford Technical Solutions Ltd, `https://www.oxts.com/app/uploads/2018/02/rtman.pdf`.

[28] H. B. Pacejka, *Tire and vehicle dynamics*, 2nd ed., Butterworth-Heinemann, 2006.

[29] P.Sushma; B.L. Rajalakshmi Samaga; K.P. Vittal, *DQ modeling of induction motor for virtual flux measurement*, IEEE 2010 Conference Proceedings IPEC, 10 2010, pp. 903–908.

[30] B. Schofield; T. Hägglund; A. Rantzer, *Vehicle dynamics control and controller allocation for rollover prevention*, IEEE International Conference on Control Applications, 11 2006, pp. 149 – 154.

[31] D. Schramm; M. Hiller; R. Bardini, *Vehicle dynamics: Modeling and Simulation*, Springer Publishing Company, Inc., 2014.

[32] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, 1st ed., Wiley-Interscience, 2006.

[33] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013.

[34] J. Valappil and Ch. Georgakis, *Systematic Estimation of State Noise Statistics for Extended Kalman Filters*, AIChE Journal **46** (2000), p. 292 – 308.

Appendix B

CD contents

DIP.pdf	**Electronic version of the Master's thesis**
img	**Images used in the thesis**
eForce_INS_Data	**Measured data from the formula**
Simulink	**Simulink models**
Nonlinear Single Track	Non-linear simulation model
Matlab	**Matlab files**
Formula	Formula data processing
KF scripts	Scripts for Kalman filtering
Model codes	Modelling scripts