Vehicle State Estimation Using the State Dependent Riccati Equation Technique

R.B.A. van Hoek

DC 2016.047

Coach: M. Alirezaei, PhD (TNO)

Supervisor: Prof.dr. H. Nijmeijer (TU/e)

Eindhoven University of Technology
Department of Mechanical Engineering
Dynamics & Control

TNO
Technical Sciences
Department Integrated Vehicle Safety

Eindhoven, August, 2016

Abstract

Vehicle state estimation is becoming ever more relevant with the increasing amount of control applications in vehicles. The most widely adopted method for non-linear estimation is the Extended Kalman Filter (EKF). However different techniques are also available. TNO has achieved promising results in stability control for braking using the State Dependent Riccati Equation (SDRE) technique. This technique can be interpreted as the non-linear extension to the well known Linear Quadratic Regulator. Its dual, the State Dependent Riccati Equation Estimator, is also studied in literature. However, it has not yet been applied to vehicle state estimation. Due to the successes achieved in vehicle control, there is an interest in studying its application for this novel application in vehicle state estimation.

The vehicle state estimator that has already been developed by TNO is based on the EKF. It is supported by a fairly large amount of additional logic. Development of a new estimator based on the SDRE technique without the support of this logic would therefore be an unfair comparison. Additionally, the state estimator used by TNO is unavailable on grounds of confidentiality. Therefore, a new vehicle state estimator is developed from the ground up, and implemented using both the EKF and SDRE technique. This allows to fairly compare the two methods, rather than the vehicle model or additional logic that are also implemented in the estimator.

This thesis will first present some theory about state estimation and provide the algorithms related to the EKF and SDRE based estimator. Their functionality and differences will be discussed using some simpler examples before being applied to the vehicle state estimation problem.

The vehicle and tyre model that are used in the vehicle state estimator are discussed step-by-step, to allow for easy reproduction. The motivation for the choice of vehicle and tyre model are given. The process of parameter identification of the developed vehicle model is described. Numerical results are used to compare the vehicle model with measurements, which demonstrate accurate state descriptions.

The implementation of this vehicle model for use in the vehicle state estimator is then described. For the EKF this describes the process of derivation of the dynamics and measurements Jacobians. However, for the SDRE based estimator this is slightly more complex, due to there being more design freedom in the selection of the non-unique factorisation of the non-linear system. Additionally, all logic that is included in the state estimator is presented.

Finally, the functionality of both the EKF and SDRE estimator are presented using both simulation and experimentation results. Simulations are used to discuss the convergence properties of the estimators for friction estimation. This is not possible in experiments due to absence of a reference measurement of this friction state. Both in the presented experiments and simulations the SDRE provides very similar estimates to those provided by the EKF, but at a lower computational cost. This makes it an attractive alternative to the currently widely adopted EKF based estimator.

Nomenclature

Acronyms

ABS	Anti-lock braking system
ARE	Algebraic Riccati equation
ATP	Automotive Testing Papenburg GmbH
DOF	Degree Of Freedom
DRE	Differential Riccati equation
EKF	Extended Kalman Filter
IMU	Inertial Measurement Unit
IVDC	Integrated vehicle dynamics control
LKF	Linear Kalman Filter
LQR	Linear Quadratic Regulator
SDC	State dependent coefficient
SDRE	State dependent Riccati equation
TNO	Dutch Organisation for Applied Scientific Research
VSE	Vehicle State Estimator

Miscellaneous

$\hat{\mathbf{x}}$	Estimates of states $\mathbf{x}$
x	italic symbols refer to scalars
$\mathbf{x}$	bold symbols refer to non-scalars
$\mathbf{I_n}$	Identity matrix of size n
$E[\quad]$	Expected value operator
$\mathbb{R}^n$	Real valued set of dimension n

System Description

$\mathbf{x}$	state vector
$\mathbf{Z(t)}$	All measurements up until time t
$\mathbf{z}$	measurement vector
$\mathbf{f(x,u)}$	vector function of non-linear system dynamics
$\mathbf{h(x,u)}$	vector function of non-linear system measurements
$\mathbf{e}$	Estimation error
$\mathbf{w}$	process noise vector
$\mathbf{v}$	measurement noise vector
$\mathbf{u}$	input vector
$\mathbf{A(x,u)}$	Linear-like structured description of non-linear system dynamics
$\mathbf{H(x,u)}$	Linear-like structured description of non-linear system measurements
$\mathbf{B}$	input matrix
$\mathbf{\Gamma}$	process noise matrix
$\mathbf{G}$	effective process variance matrix
$\mathbf{W}$	process noise variance matrix

$\mathbf{V}$	measurement noise variance matrix
$\mathbf{K}$	Filter gain matrix for measurement correction
$\mathbf{P}$	Error covariance matrix
$\mathbf{S}$	Innovation covariance matrix
$\boldsymbol{\eta}$	parametrisation variable
$\boldsymbol{\nu}$	Innovation

Tyre Parameters

$B_{x,y}$	stiffness factor magic formula
$C_{\alpha y}$	Pure slip Cornering stiffness
$C_{\kappa x}$	Pure slip Longitudinal stiffness
$C_{x,y}$	shape factor magic formula
$D_{x,y}$	peak value magic formula
$E_{x,y}$	curvature factor magic formula
$b_{ex,y}$	stiffness factor exponential tyre model
$C_{ex,y}$	limit normalised force exponential tyre model
$E_{ex,y}$	curvature factor exponential tyre model

Vehicle model

α	lateral slip angle
δ_i	steering angle at wheel i
δ_{sw}	steering wheel angle
γ_{ack}	ackerman angle
κ	longitudinal slip ratio
μ	tyre-road friction coefficient
ω_i	rotational speed or 'wheel speed' of wheel i
ρ_{air}	Air density
$\sigma_{rel_{x,y}}$	relaxation length components
$\sigma_{x,y}$	Theoretical slip components
$\tau_{x,y}$	relaxation time constants
A_{drag}	drag (frontal) surface
$a_{x,y}$	Acceleration components
$C_{d_{\text{eff}}}$	Combined drag coefficient density and drag surface
C_d	Aerodynamic drag coefficient
d_{x,y_i}	Relative wheel position components of wheel i in chassis coordinate frame
F_{drag}	Aerodynamic drag force
F_{cx,y_i}	Force components of tyre i in chassis coordinate frame
F_{x,y_i}	Force components of tyre i in wheel plane coordinate frame
g	Gravitational constant
h_{cog}	center of gravity height
h_{cop}	center of pressure height
I_p	Polar moment of inertia of wheel
I_{zz}	Mass moment of inertia
$k_{\phi_{\text{tot}}}$	total roll stiffness of vehicle
k_{ϕ_i}	roll stiffness of axle of wheel i
L	Wheelbase
L_{tw}	Track width
m	Vehicle mass
M_{ext_i}	Externally applied driving or braking torque of wheel i
M_{z_i}	Self aligning moment of tyre i
M_{zkp_i}	Moment around king pin of tyre i

R_{e_i}	Effective rolling radius of wheel i
R_{l_i}	Loaded radius of wheel i
S	Total slip
v_{cx,y_i}	Velocity components of tyre i in chassis coordinate frame
v_{sx,y_i}	Slip velocity components of tyre i in wheel plane coordinate frame
$v^*_{sx,y}$	Effective slip velocity components after relaxation
V_x	longitudinal chassis velocity
V_y	lateral chassis velocity
r	yaw velocity
λ_μ	Scaling of road friction
ϕ	roll angle
θ	pitch angle

Contents

Chapter 1

Introduction

1.1 Problem Background

Control theory has been widely adopted in the field of automotive technologies. Multiple systems have been developed such as the Anti-lock Braking System (ABS), electronic stability control and traction control. Some of these systems have even become mandatory for newly produced cars. In the past, these systems were often developed by independent manufacturers and adopted by car manufacturers. Separate control systems could therefore give conflicting control inputs [1]. Additionally, with current development in autonomous driving and advanced driver assistance systems, the number of control systems is ever increasing. This increase in complexity resulted in the need for Integrated Vehicle Dynamics Control (IVDC). IVDC aims to maximise the synergy between various actively controlled elements in the vehicle like braking, suspension, steering and powertrain, in order to improve comfort but most importantly to increase safety for people both in- and outside the vehicle.

The Integrated Vehicle Safety Department of TNO investigates the application of IVDC methods to integrate the functionality of various control systems. Main applications include collision avoidance systems with simultaneous steering and braking. IVDC is based on controlling all vehicle states. This requires all states of the vehicle to be known. Typically, the more accurate the vehicle states are known, the better the control inputs can be determined, resulting in a higher level of safety. However, in general, not all vehicle states are necessarily measurable, either due to cost-related or physical limitations. This problem leads to the use of state estimators, which reconstruct the vehicle states based on measured system outputs and a vehicle model.

1.2 Problem Statement

The most used technique for state estimation is linear Kalman filtering. This is based on linear models and is not well suited to be applied directly to non-linear systems. Vehicle dynamics are inherently non-linear, due to the non-linear behaviour of pneumatic tires. Currently TNO has adopted an Extended Kalman Filter (EKF), which is an extension to the Kalman filter to better deal with non-linear systems. This method uses linearisation around the current state estimate.

An alternative to the EKF is the State Dependent Riccati Equation (SDRE) based non-linear estimator. SDRE uses a transformation of the non-linear system to a linear-like structure. This transformation brings the non-linear system to a linear-like structure having State Dependent Coefficients (SDC's). The SDRE based estimator is widely adopted in other fields with promising results. Additionally, the SDRE technique has been very successful in the context of control as the non-linear extension of the commonly used Linear-Quadratic Regulator (LQR).

TNO has already achieved promising results applying the SDRE technique for vehicle stability

control [2]. Additionally, it is investigating application of the SDRE technique to collision avoidance systems [3]. The successful implementation of this technique for control motivates to investigate the application of the SDRE technique to the vehicle state estimation problem.

The aim of this thesis is therefore the development of an SDRE based vehicle estimator, capable of on-line state estimation, and to make a comparison of the developed SDRE state estimator with the currently adopted EKF state estimator.

1.3 Literature Survey

This section will discuss the literature available regarding the use of the State Dependent Riccati Technique used for estimation. The main topics in this preliminary literature survey are the following

- SDRE applications
- Unique properties of SDRE and how it relates to EKF
- Performance comparison of different algorithms
- Stability and Observability

1.3.1 Applications

The SDRE technique was first proposed in [4] in the context of non-linear control. Later, in [5] the SDRE based non-linear estimator was presented as the dual of the SDRE controller. The work in [6, 7, 8] continues on this topic, also specifically covering simultaneous state and parameter estimation.

Although these publications focus primarily on theoretical problems, SDRE is also adopted in real world applications as mentioned in [9]. While real world control applications are numerous and even applied to control in automotive context [2], examples of estimation problems that applied the SDRE technique seem limited to localization and positioning problems and applications for missile guidance [10, 11, 12, 13].

1.3.2 SDRE Unique Properties

The base of the SDRE technique is the transformation of the system to a linear-like structure. Consider the following non-linear system.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \Gamma\mathbf{w}(t), \qquad \mathbf{x} \in \mathbb{R}^n \tag{1.1}$$
$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}(t), \qquad \mathbf{z} \in \mathbb{R}^m \tag{1.2}$$

With $\mathbf{w}(t) \in \mathbb{R}^p$ and $\mathbf{v}(t) \in \mathbb{R}^m$, being Gaussian white noise with covariances $\mathbf{W} \in \mathbb{R}^{p\times p}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$ and $\mathbf{\Gamma} \in \mathbb{R}^{n\times p}$. Both SDRE control and SDRE estimation are based on bringing the system to the mentioned linear-like structure using factorization or extended linearisation [4, 5]. Provided the functions $\mathbf{f}(\mathbf{x})$, $\mathbf{h}(\mathbf{x})$ are continuously differentiable and $\mathbf{f}(\mathbf{0}) = \mathbf{0}$, $\mathbf{h}(\mathbf{0}) = \mathbf{0}$, the dynamics can be written in the linear-like structure with state dependent coefficients $\mathbf{A}(\mathbf{x})$, $\mathbf{H}(\mathbf{x})$.

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\mathbf{x}, \qquad \mathbf{A}(\mathbf{x}) \in \mathbb{R}^{n\times n} \tag{1.3}$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{H}(\mathbf{x})\mathbf{x}, \qquad \mathbf{H}(\mathbf{x}) \in \mathbb{R}^{m\times n} \tag{1.4}$$

For the general multi-variable system the parametrisations or factorisations $\mathbf{A}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ are non-unique, as also noted in [4, 5]. Consider now for example the function $z = x_1x_2$, for which

can be shown that two parametrisations exist

$$\mathbf{H}_1(\mathbf{x}) = \begin{bmatrix} 0 & x_2 \end{bmatrix} \quad \rightarrow \quad \mathbf{H}_1(\mathbf{x})\mathbf{x} = \begin{bmatrix} 0 & x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 x_2 = z \tag{1.5}$$

$$\mathbf{H}_2(\mathbf{x}) = \begin{bmatrix} x_1 & 0 \end{bmatrix} \quad \rightarrow \quad \mathbf{H}_2(\mathbf{x})\mathbf{x} = \begin{bmatrix} x_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 x_2 = z \tag{1.6}$$

However, since these two parametrisations exist, an infinite set of parametrisations can also be proven to exist, since

$$\mathbf{H}(\mathbf{x}, \eta)\mathbf{x} = \eta \mathbf{H}_1(\mathbf{x})\mathbf{x} + (1-\eta)\mathbf{H}_2(\mathbf{x})\mathbf{x} = \eta(x_1 x_2) + (1-\eta)(x_1 x_2) = x_1 x_2 = z \tag{1.7}$$

Where η is the parametrisation variable. This parametrisation variable is essentially used to blend multiple system descriptions. This non-uniqueness introduces additional design freedom not present in comparable algorithms like the extended Kalman filter. Applications for this additional degree of freedom are provided in [5] and include the prevention of loss of observability, avoiding singularities and enhancing filter performance. The resulting observer algorithm for the SDRE controller is

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}) + \mathbf{K}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})) \tag{1.8}$$

$$\mathbf{K} = \mathbf{P}\mathbf{H}(\boldsymbol{\eta}_2, \hat{\mathbf{x}})^T \mathbf{V}^{-1} \tag{1.9}$$

Where $\hat{\mathbf{x}}$, represent the state estimate, and η_2 denotes the parametrisation variable for the output parametrisation $\mathbf{H}(\mathbf{x})$. Note the absence of process noise, in the state estimate equation (1.8). Since the noise is assumed zero mean, the expected value $E[w(t)] = 0$. As such, the average value of noise is not included in the estimate equation. Since the SDRE estimator is the dual of the SDRE controller, in literature the error covariance matrix $\mathbf{P}$ is often found by solving the Algebraic Riccati Equation (ARE),

$$\begin{aligned} \mathbf{0} &= \mathbf{A}(\boldsymbol{\eta}_1, \hat{\mathbf{x}})\mathbf{P} + \mathbf{P}\mathbf{A}(\boldsymbol{\eta}_1, \hat{\mathbf{x}})^T - \mathbf{P}\mathbf{H}(\boldsymbol{\eta}_2, \hat{\mathbf{x}})^T \mathbf{V}^{-1} \mathbf{H}(\boldsymbol{\eta}_2, \hat{\mathbf{x}})\mathbf{P} + \mathbf{G} \\ \mathbf{G} &= \boldsymbol{\Gamma}^T \mathbf{W} \boldsymbol{\Gamma} \end{aligned} \tag{1.10}$$

Where η_1 denotes the parametrisation variable for the dynamics parametrisation $\mathbf{A}(\mathbf{x})$. In the context of control, usage of the ARE is the result of the underlying infinite horizon optimal control problem and leads to heavy computations for real time applications [14, 15]. However, also examples can be found where a numerical value for $\mathbf{P}$ is obtained using the Differential Riccati Equation (DRE) [8] (1.11), originally intended to lower the computational cost. Additionally, this method also has a closer relation to the Extended Kalman Filter and is arguably a better alternative.

$$\dot{\mathbf{P}} = \mathbf{A}(\boldsymbol{\eta}_1, \hat{\mathbf{x}})\mathbf{P} + \mathbf{P}\mathbf{A}(\boldsymbol{\eta}_1, \hat{\mathbf{x}})^T - \mathbf{P}\mathbf{H}(\boldsymbol{\eta}_2, \hat{\mathbf{x}})^T \mathbf{V}^{-1} \mathbf{H}(\boldsymbol{\eta}_2, \hat{\mathbf{x}})\mathbf{P} + \mathbf{G}, \qquad \mathbf{P}(t_0) = \mathbf{P_0} \tag{1.11}$$

The presented equations for the SDRE filter algorithm, (1.8), (1.9) and (1.11), are very similar to the equations used in the extended Kalman filter. However, the SDRE based algorithm allows more design freedom in the selection of the linear-like representation of the systems dynamics $\mathbf{A}(\mathbf{x})$ and measurements $\mathbf{H}(\mathbf{x})$. This unique property of the SDRE technique could potentially prove beneficial when applied to vehicle state estimation.

1.3.3 Performance Comparison

Due to the similarities between the SDRE and Kalman algorithms, many comparisons have been made between the two filters. The SDRE filter is first introduced in [5]. It compares the infinite horizon steady-state SDRE filter with both the infinite horizon steady-state linearised Kalman filter and the extended Kalman filter. The infinite horizon steady-state label refers here to the usage of the Algebraic Riccati Equation (ARE) rather than integrating the Differential Riccati Equation (DRE). The comparison is made with a simple two dimensional, undamped, pendulum problem. Although the paper mentions the additional degree of freedom available in the SDRE filter, it is not used in this example. With the single factorisation that is presented for the SDRE filter, a direct analytical solution to the Kalman gain is found. For real time applications this is beneficial, since the ARE does not need to be solved on-line. The same holds for the linearised Kalman filter. The extended Kalman filter does not allow for such a solution since the DRE is integrated on-line. In contrast to [5], which uses the ARE, a different form is presented in [8],[12] and [16], where the SDRE filter is implemented by solving the DRE.

Further comparisons are made in [10], now specifically addressing the additional degree of freedom that is present in the State Dependent Coefficients (SDC's) resulting from factorisation. A simple example problem is proposed in [5], for which numerical results are presented in [10].

Finally, [12] is considered, in which a number of of example problems are presented. Parameter estimation of a constant frequency of oscillation, simultaneous state and parameter estimation for a falling object under influence of non constant drag coefficient and finally, tracking of a point mass describing a parabola with non-linear measurements. Interestingly, a different approach is used than in the previously discussed papers. The difference being that the process noise is tuned separately for the SDRE and EKF algorithms. This results in a comparison of the most optimal cases for both algorithms rather than using identical parameters for both filters as is done in the papers mentioned earlier. Additionally, [12] presents a theoretical value for the expected error. Although it is not mentioned explicitly in the paper, this is most likely derived from the calculated error covariance matrix $\mathbf{P}$. A similar approach is also used in [17] in the context of evaluating the EKF performance. In the latter however, the innovation $(\hat{\mathbf{z}} - \mathbf{z})$ covariance is used, as typically true states are not available in the context of estimators.

1.3.4 Stability and Observability

Stability of non-linear systems is notoriously difficult to prove. For discrete time Extended Kalman filters, this is discussed in both [18], and [19], which show convergence under very strict criteria. In the context of SDRE algorithm similar attempts have been made. Paper [4] gives stability and optimality criteria for the SDRE algorithm in the context of control. Paper [8] assesses the controllability of the SDRE algorithm using Lyapunov's stability [20] theorem and remarks the difficulties in proving stability in the presence of both $\hat{\mathbf{x}}$ and $\mathbf{x}$. However, this proof for control stability is incomplete as not all terms in the derivative of the energy function are considered. Paper [11] attempts a similar approach in the context of an observer for a localization problem. The suggested Lyapunov function is taken directly from [8], where it was used to prove stability of a SDRE controller around the origin. However, in [11] the problem is not to control the system to the origin, but rather one of estimation. This makes the suggested Lyapunov function incorrect. [16] discusses exponential stability for an alternative to the SDDRE observer. Robust parametrisation for control applications is presented in [21].

Observability criteria for non-linear systems are presented in [8, 22]. Algorithmic observability is briefly mentioned in [10]. Algorithmic observability means that the matrices used in the estimator to represent the system in a linear-like structure (eg. jacobians or SDC's) are observable in the linear sense, such that they pass the linear rank criterion. When this criterion is not satisfied, the filter can encounter numerical issues, resulting in instability. Criteria for this algorithmic

observability are also given in [12, 16, 15]. However, it is also demonstrated that even though the criteria are met, the filter might diverge or convergence when criteria are not met.

1.4 Project Goals

As described in the problem statement the aim of this work is applying the SDRE technique in the context of Vehicle State Estimation. Specifically, a comparison between the currently adopted method of EKF and the newly suggested SDRE technique is of interest. Due to confidentiality, the TNO Vehicle State Estimator(VSE) is unavailable for thorough evaluation. State estimates of the TNO VSE are only available for a limited set of measurements. This implies that a new benchmark observer based on the currently adopted EKF method is required. The vehicle state estimator requires the development of a non-linear vehicle model. This model will be validated using measurement data from the Toyota Prius test vehicle from TNO. A method for fair comparison and evaluation of the developed estimators is required. The resulting SDRE based estimator should be compared with respect to the currently adopted EKF based estimator and, where possible, the TNO VSE. This comparison should include both fundamental theoretical background and numerical results. Stability and optimality proofs are difficult as mentioned in the literature survey. Emphasis during this thesis will therefore be on the practical implementation. Both simulation data, where true states are available, and measured data from an experimental set-up will form the basis of evaluation of both algorithms.

1.5 Report Structure

The structure of the report is as follows. First theory regarding state estimation will be discussed in Chapter 2. Both EKF and SDRE filters will presented in full, after which a fundamental comparison is made between the two. Next, in Chapter 3, the model used in the vehicle state estimator will be presented together with motivation for its selection. Chapter 4 will then cover all additional steps that are required to implement the vehicle model for usage in an observer. In Chapter 5, a model will be used to generate datasets on which to test the VSE. Specific attention is given to cases that cannot be tested on the experimental set-up. Chapter 6 then discusses implementation of the VSE for the experimental set-up, together with the obtained results. Finally, in Chapter 7, the conclusions and recommendations will be presented.

Chapter 2

Estimation Theory

In control applications, promising results can be achieved using full state feedback. This requires the states of the system to be known. System states need not all be measurable in general, either due to physical limitations or cost restrictions. Lack of state information demands for a state reconstructing algorithm. These algorithms combine a dynamic model with measured outputs to reconstruct the states, which is illustrated schematically in Figure 2.1. These algorithms are referred to as observers for the deterministic case, estimators for the stochastic case or, when an input is absent, filters. This chapter will cover the theory regarding these algorithms and present the equations related to some of the more common algorithms. It will also present a comparison with comprehensive examples.

2.1 Estimators

Estimators require a description of the dynamics and measurements of the system. Process and measurement noise are both assumed to be additive. This additivity assumption simply implies that noise is only added to the dynamics and measurements rather than corrupting dynamics and measurement directly inside the functions $\mathbf{f}(\mathbf{x}, \mathbf{u}(t))$ and $\mathbf{h}(\mathbf{x}, \mathbf{u}(t))$. Under this assumption of additive noise, the general non-linear time-invariant system can be described by the following equations

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{\Gamma}(\mathbf{x})\mathbf{w}(t), & \mathbf{x} &\in \mathbb{R}^n \\ \mathbf{z}(t) &= \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{v}(t), & \mathbf{z} &\in \mathbb{R}^m \end{aligned} \tag{2.1}$$

In which $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathbb{R}^n$ and $\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathbb{R}^m$ are the vector valued functions of the non-linear system dynamics and measurements, $\mathbf{x}(t) \in \mathbb{R}^n$ the state vector, $\mathbf{z}(t) \in \mathbb{R}^m$ the output vector, $\mathbf{u}(t) \in \mathbb{R}^q$ the input vector, $\mathbf{\Gamma} \in \mathbb{R}^{n \times p}$, $\mathbf{v}(t) \in \mathbb{R}^m$ and $\mathbf{w}(t) \in \mathbb{R}^p$ are zero mean Gaussian noise vectors with the following properties

$$E[\mathbf{w}(t)\mathbf{w}^T(t+\tau)] = \mathbf{W}(t)\delta(\tau) \tag{2.2}$$

$$E[\mathbf{v}(t)\mathbf{v}^T(t+\tau)] = \mathbf{V}(t)\delta(\tau) \tag{2.3}$$

Here $E[\ ]$ denotes the expected value operator. $\mathbf{W} \in \mathbb{R}^{p \times p}$ and $\mathbf{V} \in \mathbb{R}^{m \times m}$ represent the covariance matrices for process and measurement noise respectively. See appendix A.1 for further explanation.

Different conditions on the optimality of the state estimate can be chosen. Four estimate criteria are distinguished in [17], the maximum likelihood, the maximum a posteriori, least squares and minimum mean-squared error.

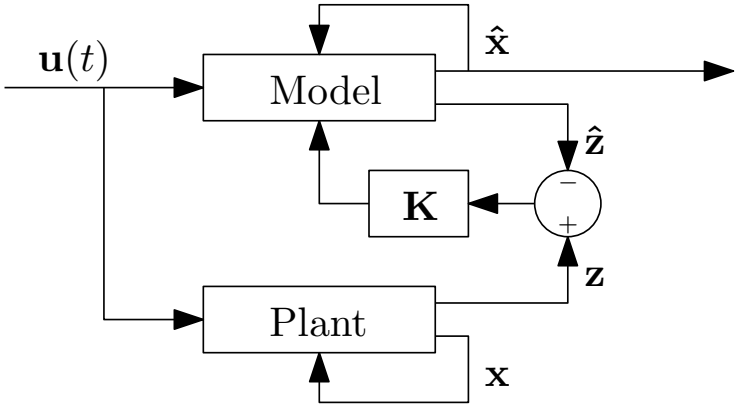


Figure 2.1: Basic functionality of an observer, states are estimated based on the dynamic model and corrected by output comparisons.

$$\hat{\mathbf{x}}^{ML}(t) \triangleq \arg\max_{\mathbf{x}} E[\mathbf{Z}(t) \mid \mathbf{x}(t)] \tag{2.4}$$

$$\hat{\mathbf{x}}^{MAP}(t) \triangleq \arg\max_{\mathbf{x}} E[\mathbf{x}(t) \mid \mathbf{Z}(t)] \tag{2.5}$$

$$\hat{\mathbf{x}}^{LS}(t) \triangleq \arg\min_{\mathbf{x}} \int_0^t [\mathbf{z}(\tau) - \mathbf{h}(\mathbf{x}(\tau))]^{\mathbf{T}} [\mathbf{z}(\tau) - \mathbf{h}(\mathbf{x}(\tau))]\, d\tau \tag{2.6}$$

$$\hat{\mathbf{x}}^{MMSE}(t) \triangleq \arg\min_{\mathbf{x}} E[\mathbf{e}^T\mathbf{e} \mid \mathbf{Z}(t)] \tag{2.7}$$

$$\mathbf{e}(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t) \tag{2.8}$$

Here $\mathbf{Z}(t)$ denotes all measurements until time t. The Maximum Likelihood principle is mostly used in finding constant (non-random) variables. In cases where parameters are considered to be a random variable, with known probability density function, Bayes Theorem (see appendix A.1) can be used to derive the Maximum A Posteriori estimate. For constant parameters and noise corrupted measurements the Least Squares method is used. Finally, as counterpart of the Least Squares method for random variables the Minimum Mean Squared Error method is used. Relations between these estimates are presented in [17].

2.1.1 Recursive Filters

The equations (2.4)−(2.7) show that state estimates are conditioned on all previous measurements, $\mathbf{Z}(t)$. It can be understood that when implemented in an algorithm, the computational effort increases for increasing amount of samples that have been taken from the system. Clearly, for online estimators this is unacceptable since at some point the computational load will become too large to run real-time.

Fortunately, most algorithms can be re-written in a recursive form. This recursive form uses only the estimate of the last step in combination with only the newest observation, rather than all previous observations. All previous information is summarised in the previous state estimate and the error covariance.

2.1.2 State Augmentation

System parameters are not necessarily fully known, or they can change in time. Unless accounted for, this could lead to large errors in the estimation of the states. In these cases the state description can be described such that the parameter is appended to the existing states as an

augmented state with its own process noise $\mathbf{w}_p(t)$ such that

$$\begin{bmatrix}\dot{\mathbf{x}}\\ \dot{\mathbf{x}}_p\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\mathbf{x})\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{w}(t)\\ \mathbf{w}_p(t)\end{bmatrix} \tag{2.9}$$

This method is also described in [6, 12]. This augmentation of the state space formulation is not only helpful to decrease the errors and improve convergence of the filters, but the estimate of this parameter can also be used to improve the accuracy of control applications. This is not limited to the full state feedback mentioned earlier, but can also improve feed forward systems and planning of a (valid) trajectory in reference generators.

2.1.3 Linear Kalman Filter (LKF)

For linear systems the description of (2.1) collapses to a linear state space formulation given by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}(t) + \mathbf{\Gamma}\mathbf{w}(t)\\ \mathbf{z} &= \mathbf{H}\mathbf{x} + \mathbf{v}(t)\end{aligned} \tag{2.10}$$

Where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{z}(t) \in \mathbb{R}^m$, $\mathbf{u}(t) \in \mathbb{R}^q$, $\mathbf{w}(t) \in \mathbb{R}^p$, $\mathbf{v}(t) \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{H} \in \mathbb{R}^{m\times n}$,$\mathbf{B} \in \mathbb{R}^{n\times q}$ and $\mathbf{\Gamma} \in \mathbb{R}^{n\times p}$. The linear Kalman filter is comprehensively summarised in [23] and [24]. A more thorough background is provided in both [17] and [25]. The Kalman filter provides a conditional mean estimate of the true state, given by (2.11).

$$\hat{\mathbf{x}}(t) = E[\mathbf{x}(t) \mid \mathbf{z}(t)] \tag{2.11}$$

$$\hat{\mathbf{x}}(t) = \arg\min_{\mathbf{x}} E[\mathbf{e}(t)^T\mathbf{e}(t) \mid \mathbf{z}(t)] \tag{2.12}$$

Which is shown in [17] to be identical to the estimate of (2.12). It does so in a prediction (*a priori*) step followed by a measurement correction (*a posteriori*) step. In discrete time these are identified as two separate steps (see also appendix A.4). But as derived in [25] for continuous time, both steps are combined in a single equation (2.13).

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u}(t) + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{2.13}$$

The term $\mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u}(t)$ can be interpreted as the *a priori* term, projecting the state estimate ahead in time, under the influence of input $\mathbf{u}(t)$ and dynamics $\mathbf{A}$. The $\mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}})$ term is then the *a posteriori* correction term. This uses the *innovation*, $\boldsymbol{\nu} = \boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}}$, to correct the state estimate. Essential in this step, is the Kalman gain $\mathbf{K}$, which translates the innovation to a correction in the state estimate.

The Kalman gain is calculated using equation (2.14). The error covariance matrix $\mathbf{P}$ describes the covariance of the error in the state estimate and is, in essence, a measure for the uncertainty in the current estimate. Propagation of this uncertainty through the linear system follows the Differential Riccati Equation (DRE) (2.15), as presented in [25]. The term $(\mathbf{AP} + \mathbf{PA^T})$ accounts for propagation of uncertainty through the dynamics of the system and the term $\mathbf{G}$ accounts for additive process noise. The observations of the systems reduce the uncertainty via $\mathbf{PH^TV^{-1}HP}$, wherein the measurement noise variance matrix $\mathbf{V}$, quantifies the intensity of measurement noise $\mathbf{v}(t)$.

$$\mathbf{K} = \mathbf{PH^TV^{-1}} \tag{2.14}$$

$$\dot{\mathbf{P}} = \mathbf{AP} + \mathbf{PA^T} - \mathbf{PH^TV^{-1}HP} + \mathbf{G}, \qquad \mathbf{P}(t_0) = \mathbf{P}_0 \tag{2.15}$$

$$\mathbf{G} = \mathbf{\Gamma^T W \Gamma} \tag{2.16}$$

For linear systems the error covariance rapidly converges to a fixed value independent of initial condition [17, 25]. This allows for calculation for the steady-state Kalman gain off-line. This is done by solving the algebraic Riccati equation for $\mathbf{P}$ (2.17). This reduces the online calculation of the filter significantly, as $\mathbf{P}$ and $\mathbf{K}$ are now only calculated once instead of every time step.

$$\mathbf{AP} + \mathbf{PA^T} - \mathbf{PH^T V^{-1} HP} + \mathbf{G} = \mathbf{0} \tag{2.17}$$

However, in reality systems are hardly ever truly linear. Hence, the LKF algorithm can no longer be applied. An extension to the Kalman filter is introduced to overcome this problem, which is the Extended Kalman Filter [17] [25].

2.1.4 Extended Kalman Filter (EKF)

The EKF [17, 25, 23] is based on linearisation of the system dynamics and observations at the current state estimate. In this case Jacobian matrices $\mathbf{A}(\mathbf{x}, \mathbf{u})$ and $\mathbf{H}(\mathbf{x}, \mathbf{u})$ are constructed using

$$\mathbf{A_{[i,j]}} = \left.\frac{\partial \mathbf{f}_{[i]}}{\partial \mathbf{x}_{[j]}}\right|_{(\hat{\mathbf{x}},\mathbf{u})} \tag{2.18}$$

$$\mathbf{H_{[i,j]}} = \left.\frac{\partial \mathbf{h}_{[i]}}{\partial \mathbf{x}_{[j]}}\right|_{(\hat{\mathbf{x}},\mathbf{u})} \tag{2.19}$$

Updating the state estimate is still of similar structure as in the case of the LKF. Equation (2.13) now transforms into (2.20). Note that the full non-linear model $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ is used to describe the dynamics of the system. The Jacobians $\mathbf{A}(\hat{\mathbf{x}}, \mathbf{u})$ and $\mathbf{H}(\hat{\mathbf{x}}, \mathbf{u})$ are used in equation (2.21) to integrate the error covariance approximation forward in time. It should be noted that matrix $\mathbf{P}$ is no longer the error covariance matrix, but only an approximation. Appendix A.3 provides further explanation on this subject. Note that since $\mathbf{A}(\hat{\mathbf{x}}, \mathbf{u})$ and $\mathbf{H}(\hat{\mathbf{x}}, \mathbf{u})$ are now no longer constant in time, it is not possible to calculate a steady-state value off-line as is the case for the linear Kalman filter. The state estimates and error covariance matrix are now calculated by

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}(t)) + \mathbf{K}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})), \qquad \hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0 \tag{2.20}$$

$$\dot{\mathbf{P}} = \mathbf{A}(\hat{\mathbf{x}})\mathbf{P} + \mathbf{PA^T}(\hat{\mathbf{x}}) - \mathbf{PH^T}(\hat{\mathbf{x}})\mathbf{V^{-1}H}(\hat{\mathbf{x}})\mathbf{P} + \mathbf{G}, \qquad \mathbf{P}(t_0) = \mathbf{P}_0 \tag{2.21}$$

2.1.5 State Dependent Riccati Equation (SDRE)

Another estimation technique for non-linear systems is the State Dependent Riccati Equation Filter (SDREF). It functions by use of factorisation [5, 10, 8, 7], rather than linearisation as is the case for EKF. The factorisation $\mathbf{A}(\mathbf{x})$ of $\mathbf{f}(\mathbf{x})$ can be found on the following conditions. Provided that $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{f}(\mathbf{x})$ being continuous, a continuous matrix valued function $\mathbf{A}(\mathbf{x})$ can always be found satisfying (2.22).

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\mathbf{x} \tag{2.22}$$

This is used to brings the dynamics given to the form below, with state dependent coefficients (SDC's) $\mathbf{A}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})\mathbf{u}(t) + \mathbf{\Gamma}(\mathbf{x})\mathbf{w}(t) \tag{2.23}$$

$$\mathbf{z} = \mathbf{H}(\mathbf{x})\mathbf{x} + \mathbf{v}(t) \tag{2.24}$$

The factorisation given by $\mathbf{A}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ is non-unique in case of multiple states as demonstrated in section 1.3 and [4, 5, 10]. This property gives extra degrees of freedom and can be

utilised to avoid singularities, loss of observability [4, 12, 11] or enhance filter performance [10]. An example was already presented in section 1.3. This additional degree of freedom results in an infinite set of factorisations being available according to (2.25) and (2.26),

$$\mathbf{A}(\mathbf{x}, \eta_1) = \eta_1 \mathbf{A_1}(\mathbf{x}) + (1 - \eta_1)\mathbf{A_2}(\mathbf{x}) \tag{2.25}$$

$$\mathbf{H}(\mathbf{x}, \eta_2) = \eta_2 \mathbf{H_1}(\mathbf{x}) + (1 - \eta_2)\mathbf{H_2}(\mathbf{x}) \tag{2.26}$$

where $\mathbf{A_1}(\mathbf{x})$ and $\mathbf{A_2}(\mathbf{x})$ are two possible factorisations, both satisfying (2.22), and $\eta_1, \eta_2 \in [0, 1]$. The estimator update equation then becomes

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}(t)) + \mathbf{K}(\mathbf{z} - \mathbf{H}(\hat{\mathbf{x}}, \eta_2)\hat{\mathbf{x}}) \qquad \hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0 \tag{2.27}$$

The SDRE algorithm further resembles the extended Kalman filter algorithms, using (2.21) and (2.14) to calculate the Kalman gain matrix. However, in literature the error covariance is also often calculated using the (steady-state solution) Algebraic Riccati Equation (ARE). This is due to the SDRE filter being the dual of the SDRE controller.

$$\mathbf{A}(\hat{\mathbf{x}}, \eta_1)\mathbf{P} + \mathbf{P}\mathbf{A}^{\mathbf{T}}(\hat{\mathbf{x}}, \eta_1) - \mathbf{P}\mathbf{H}^{\mathbf{T}}(\hat{\mathbf{x}}, \eta_2)\mathbf{V}^{-1}\mathbf{H}(\hat{\mathbf{x}}, \eta_2)\mathbf{P} + \mathbf{G} = \mathbf{0} \tag{2.28}$$

It should be noted that (2.28) strongly resembles (2.17). However they are not identical due to the state dependent arguments in (2.28). Its relevance can be argued, since solving the equation at a given time step will not provide a steady-state solution like is the case for (2.17). For the ARE to have solutions the system should also be observable in the linear sense, satisfying (A.20).

For the estimation problem, the solution of the Riccati equation represents the error covariance. Therefore, it is possible to use the DRE (2.29) to update the error covariance matrix $\mathbf{P}$ similarly to the EKF algorithm, with a significant reduction in computation cost as also demonstrated in [8]. Additionally, the solution for the DRE does not require the system to be observable in the linear sense, which ensures that filter gains are still calculated, even when the pair $[\mathbf{A}(\hat{\mathbf{x}}, \eta_1), \mathbf{H}(\hat{\mathbf{x}}, \eta_2)]$ is not observable in the linear sense at a given state.

$$\dot{\mathbf{P}} = \mathbf{A}(\hat{\mathbf{x}}, \eta_1)\mathbf{P} + \mathbf{P}\mathbf{A}^{\mathbf{T}}(\hat{\mathbf{x}}, \eta_1) - \mathbf{P}\mathbf{H}^{\mathbf{T}}(\hat{\mathbf{x}}, \eta_2)\mathbf{V}^{-1}\mathbf{H}(\hat{\mathbf{x}}, \eta_2)\mathbf{P} + \mathbf{G}, \qquad \mathbf{P}(t_0) = \mathbf{P}_0 \tag{2.29}$$

Potential benefits of the SDC parametrisation used in the SDRE filter include the additional degrees of freedom in estimator design. Additionally, the SDC parametrisation does not require to explicitly derive the Jacobian of the non-linear system.

2.2 Literature Discussion

With the theory discussed in Section 2.1, it is possible to demonstrate the estimators using some simple numerical examples. This section will discuss some of the results presented in the literature summarised in Section 1.3. Emphasis is placed on the comparison between the estimator algorithms. A simple pendulum system will first be presented, after which a system with loss of observability is also covered.

2.2.1 Pendulum

First a simple pendulum system will be considered. An undamped pendulum system is proposed in [5], with a measurement of the acceleration of the pendulums bob. This results in the following set of equations of motion and output map

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{L}\sin x_1 \\ z &= -\frac{g}{L}\sin x_1 \end{aligned} \tag{2.30}$$

The factorisations used in the SDRE filter are also chosen identical to that of the paper

$$\mathbf{A}(\hat{\mathbf{x}}) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L}\frac{\sin\hat{x}_1}{\hat{x}_1} & 0 \end{bmatrix}, \qquad \mathbf{H}(\hat{\mathbf{x}}) = \begin{bmatrix} -\frac{g}{L}\frac{\sin\hat{x}_1}{\hat{x}_1} & 0 \end{bmatrix} \tag{2.31}$$

Also identical to the paper is the usage of the ARE for finding the error covariance approximation $\mathbf{P}$. Noise intensities are chosen identical to that of the paper, such that $\mathbf{V} = 2$, $G_{11} = 0.05$, $G_{22} = 0.05$. A fourth order runge-kutta solver with a fixed step size of $\Delta t = 0.005$ is used. For the Extended Kalman filter an initial error covariance of zero is used, also identical to that of the paper. Two cases are supported with numerical results in the paper [5], which compare a correct and an incorrect initial estimate respectively. The results for a correct and incorrect initial estimate can be found in respectively Figure 2.2 and Figure 2.3.

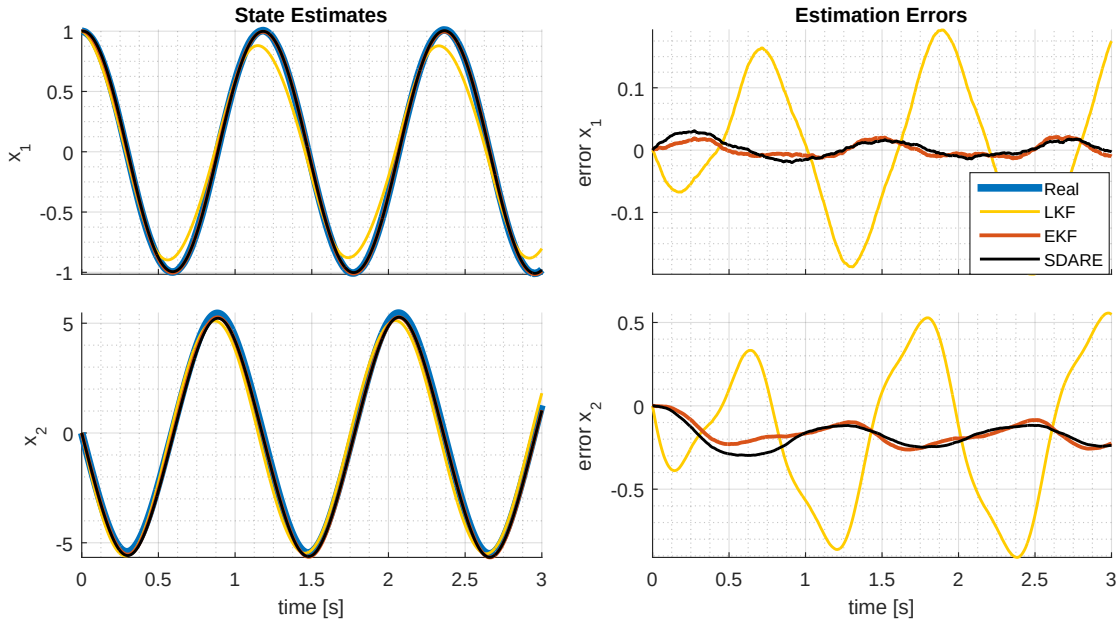


Figure 2.2: Comparison of algorithms for simple pendulum system as presented in [5], initial estimate is equal to initial condition. $P_{11} = 0$ and $P_{22} = 0$ for EKF, L=0.3048m, g=9.81

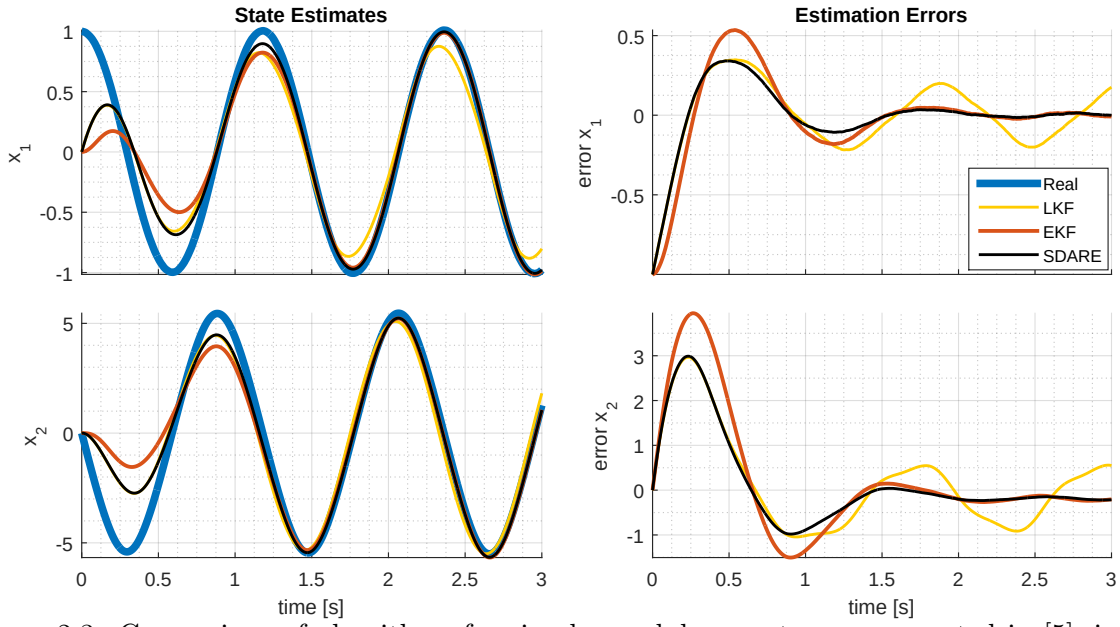


Figure 2.3: Comparison of algorithms for simple pendulum system as presented in [5], initial estimate is not equal to initial condition. $P_{11} = 0$ and $P_{22} = 0$ for EKF, L=0.3048m, g=9.81

The results are comparable with that of the paper, showing near identical behaviour for EKF and SDRE for the correct initial condition. As is to be expected the linear nature of the LKF is not capable of providing accurate non-linear estimates, as seen by the larger estimation errors. An important point of discussion is the case where the initial estimate is not identical to that of the initial condition. As is the case for the results presented in [5], the LKF and SDRE show similar initial growth rates, and EKF and SDRE show very similar final estimates. Note that SDRE is indicated in the legend as SDARE, referring to the usage of the ARE for calculating the error covariance matrix. Although not directly stated in the paper, the fast convergence and small final error present SDRE as the best performing filter.

However, it could be argued that the initial condition for the error covariance in the EKF filter is a poor guess, since it indicates perfect knowledge of the initial state. This is clearly not the case in the presented problem, making it an unfair comparison. Even with a very conservative choice for P_0, the performance of the EKF estimator is significantly increased. By selecting $P_{11} = 0.05$ and $P_{22} = 0.05$, a standard deviation of $\sigma_{x_{1,2}} = 0.22$ is implied for the error in both states. The results can be found in Figure 2.4. With the updated initial condition for the error covariance matrix, the EKF clearly outperforms the LKF and SDRE estimators, in terms of convergence speed and gives identical steady-state estimates to that of the SDRE estimator.

This comparison highlight the importance of selecting a representative initial condition for the error covariance matrix. Additionally, an advantage of the SDRE estimator with the ARE also becomes evident, since it functions without this initial condition. As such, it does not suffer from a poor initial value for this error covariance matrix. However, as a consequence, the estimate provided by the SDRE is also never the optimal estimate.

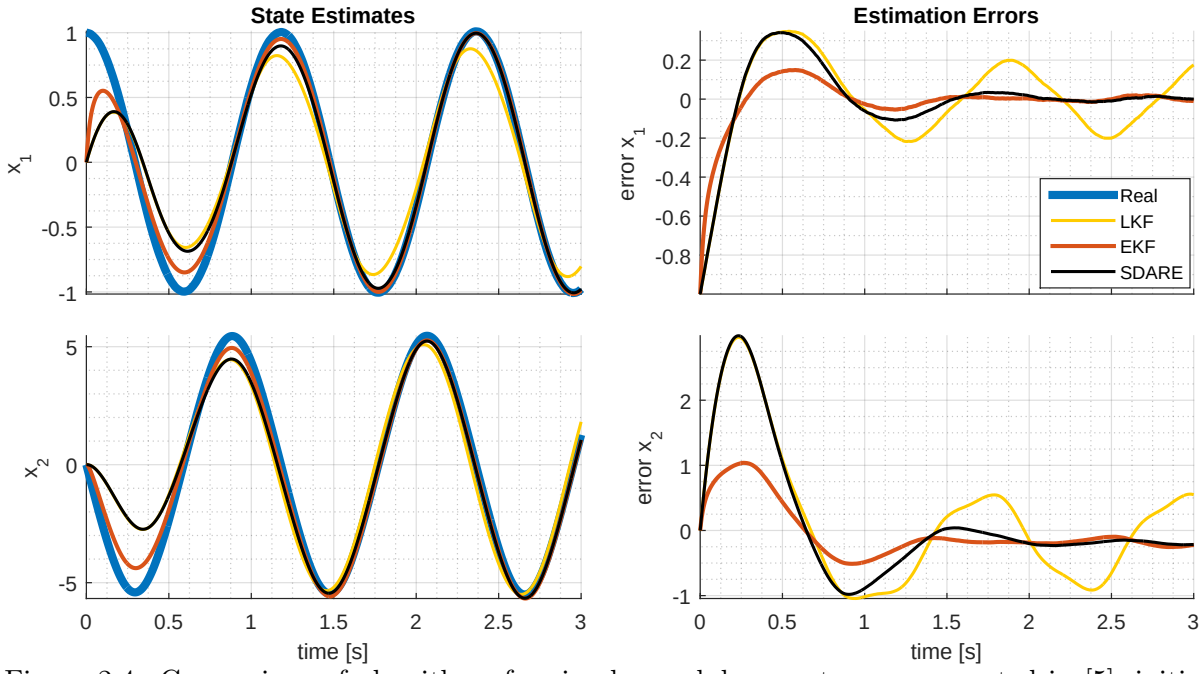


Figure 2.4: Comparison of algorithms for simple pendulum system as presented in [5], initial estimate is not equal to initial condition. $P_{11} = 0.05$ and $P_{22} = 0.05$ for EKF, L=0.3048m, g=9.81

In addition to the paper, the SDRE estimator is also implemented using the DRE. These results can be seen in Figure 2.5, where SDDRE refers to the usage of the DRE. The results demonstrate that EKF and DRE based SDRE estimator have near identical performance. This illustrates the different properties of the estimators using the algebraic and differential Riccati equation.

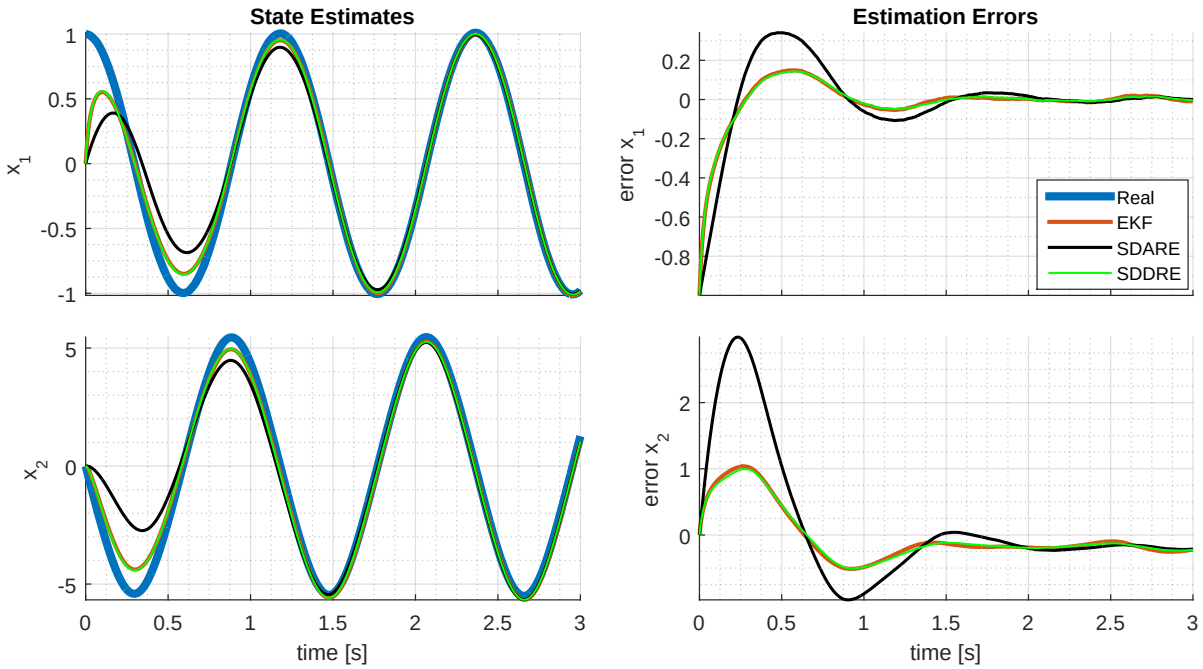


Figure 2.5: Comparison of algorithms for simple pendulum system as presented in [5], initial estimate is not equal to initial condition. $P_{11} = 0.05$ and $P_{22} = 0.05$ for EKF and SDDRE , L=0.3048m, g=9.81

2.2.2 Loss of Observability

To demonstrate the potential benefit of the additional design freedom in the SDRE filter, an example problem with loss of observability is used. The additional design freedom can be used to prevent observability loss, which will be demonstrated for the simple problem that is proposed in [5] and for which numerical results are presented in [10]. The dynamics and outputs of this system are given by

$$\begin{aligned} \dot{x_1}(t) &= w_1(t) \\ \dot{x_2}(t) &= w_2(t) \\ z_1(t) &= x_1(t) + x_2(t) + v_1(t) \\ z_2(t) &= x_1(t)x_2(t) + v_2(t) \end{aligned} \tag{2.32}$$

With $v(t)$ and $w(t)$ being zero mean, Gaussian measurement and process noise. The resulting measurement Jacobian used in the extended Kalman filter algorithm is given by

$$\frac{\partial \mathbf{h}(\hat{\mathbf{x}})}{\partial \mathbf{x}}(\hat{\mathbf{x}}) = \begin{bmatrix} 1 & 1 \\ \hat{x}_1 & \hat{x}_2 \end{bmatrix} \tag{2.33}$$

It can directly be seen that (2.33) loses rank whenever $\hat{x}_1 = \hat{x}_2$, resulting in observability rank deficiency. For the SDC parametrisation however, there exist two distinct parametrisations which are both equally valid. As presented in (2.26) and (1.5), these parametrisations can be combined into a single parametrisation with an addition degree of freedom as given by

$$\mathbf{H_1}(\hat{\mathbf{x}}) = \begin{bmatrix} 1 & 1 \\ 0 & \hat{x}_2 \end{bmatrix}, \qquad \mathbf{H_2}(\hat{\mathbf{x}}) = \begin{bmatrix} 1 & 1 \\ \hat{x}_1 & 0 \end{bmatrix}, \qquad \mathbf{H}(\hat{\mathbf{x}}, \eta_2) = \begin{bmatrix} 1 & 1 \\ \eta_2 \hat{x}_1 & (1-\eta_2)\hat{x}_2 \end{bmatrix} \tag{2.34}$$

The additional degree of freedom created by η_2 can be used to prevent the loss of rank encountered by the EKF. In the numerical results presented by [10], a fixed value for η_2 is used. However,

this does not solve the observability issue since loss of observability is still encountered whenever $(1-\eta_2)\hat{x_2} = \eta_2\hat{x_1}$. A better choice would be to make η_2 state dependent resulting in $\eta_2(\hat{\mathbf{x}})$. A possible method of doing so is presented as

$$\eta_2(\hat{\mathbf{x}}) = \begin{cases} 0.5 & \text{if } |\hat{x}_1| > 0.1 \wedge |\hat{x}_2| > 0.1 \\ 1 & \text{if } |\hat{x}_1| \leq 0.1 \wedge |\hat{x}_2| \leq 0.1 \wedge |\hat{x}_1| < |\hat{x}_2| \\ 0 & \text{if } |\hat{x}_1| \leq 0.1 \wedge |\hat{x}_2| \leq 0.1 \wedge |\hat{x}_1| \geq |\hat{x}_2| \end{cases} \tag{2.35}$$

This formulation for $\eta_2(\hat{\mathbf{x}})$ essentially sets η_2 to 0.5 except for the cases where $\hat{x_1}$ or $\hat{x_2}$ approaches zero. It will then become 0 or 1 based on which of the states is closer to zero.

This approach is implemented in MATLAB Simulink using identical parameters used as in [10] with exception of the choice of η_2 and initial variance. $x_1(0) = 0.1$, $x_1(0) = 0.3$, $\mathbf{P}_0 = 1.5\mathbf{I}_{2x2}$, $\mathbf{W} = \mathbf{I}_{2x2}$, $\mathbf{V} = 0.1\mathbf{I}_{2x2}$, the simulation is performed in Simulink using a discrete time solver with a fixed sample frequency of 100 Hz. A Monte Carlo simulation is performed with a random initial state estimate generated using the initial error covariance matrix $\mathbf{P}_0$.

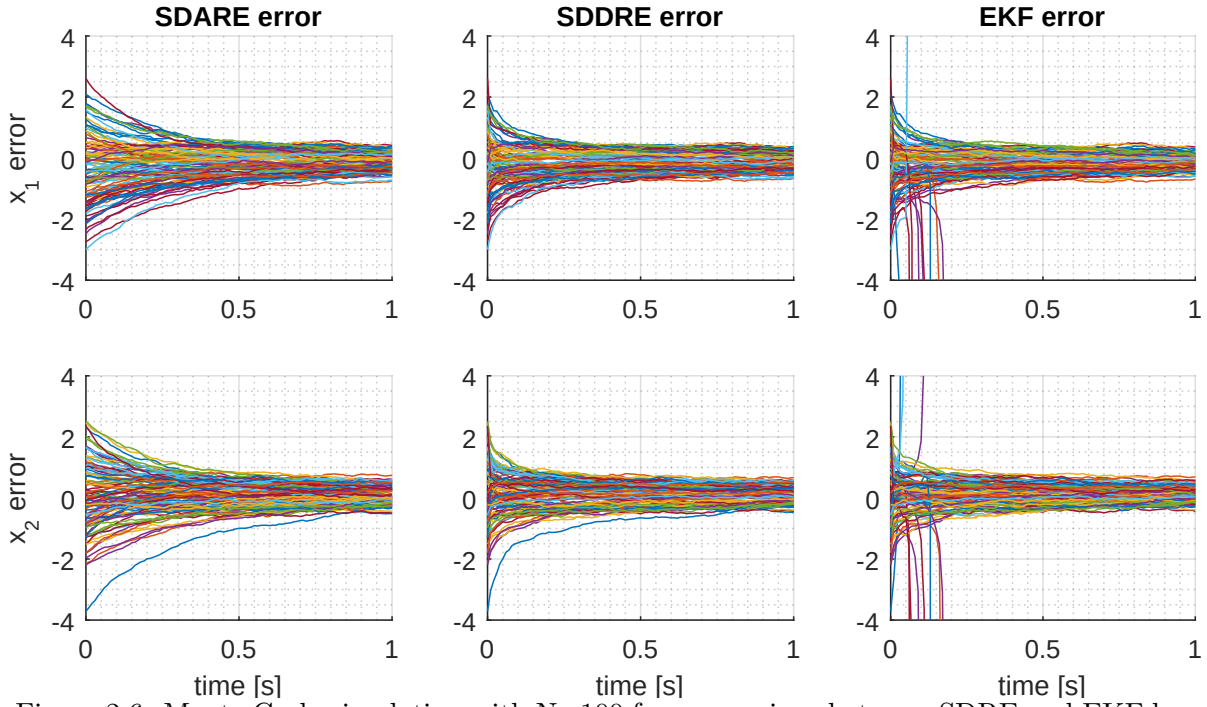


Figure 2.6: Monte Carlo simulation with N=100 for comparison between SDRE and EKF based estimation, Both ARE and DRE are implemented for SDRE.

From Figure 2.6, it can be seen that the additional degree of freedom can be used to prevent an exploding filter in case of observability loss. In contrast to [10], both the ARE and DRE are implemented, which display similar behaviour as can be seen in the figure. It should also be noted that with $\mathbf{P}_0 = \mathbf{I}_{2x2}$, far fewer diverging estimates are encountered than can be seen from the figure presented in [10]. Different solvers have been used in an attempt to reproduce the results more closely, but without success. Despite not obtaining the same results as presented in the paper, this example does clearly illustrate the potential benefits of the SDRE factorisation over EKF.

2.3 Estimator Comparison

This section will further discuss the fundamental differences between the filter algorithms introduced in section 2.1, that are not yet outlined in literature. The various steps in the algorithm will be discussed separately as to better indicate the differences. Recall the equations given in Section 2.1

$$\dot{\hat{\mathbf{x}}} = \overbrace{\mathbf{f}(\hat{\mathbf{x}},\mathbf{u})}^{\text{update}} + \overbrace{\mathbf{K}(\mathbf{z}-\mathbf{h}(\hat{\mathbf{x}},\mathbf{u}))}^{\text{correction}} \tag{2.36}$$

$$\dot{\mathbf{P}} = \overbrace{\mathbf{A}(\hat{\mathbf{x}},\mathbf{u})\mathbf{P}+\mathbf{P}\mathbf{A}(\hat{\mathbf{x}},\mathbf{u})^{\mathbf{T}}+\mathbf{G}}^{\text{update}} - \overbrace{\mathbf{P}\mathbf{H}(\hat{\mathbf{x}},\mathbf{u})^{\mathbf{T}}\mathbf{V}^{-1}\mathbf{H}(\hat{\mathbf{x}},\mathbf{u})\mathbf{P}}^{\text{correction}} \tag{2.37}$$

$$\mathbf{K} = \mathbf{P}\mathbf{H}(\hat{\mathbf{x}},\mathbf{u})^{\mathbf{T}}\mathbf{V}^{-1} \tag{2.38}$$

Observe that in (2.36), the only difference between algorithms is the calculated Kalman gain $\mathbf{K}$, as predicted output $\mathbf{h}(\hat{\mathbf{x}},\mathbf{u})$, and dynamics of $\mathbf{f}(\hat{\mathbf{x}},\mathbf{u})$ are independent of the filter algorithm. In both (2.37) and (2.38) the major difference between algorithms can be found in the method that is used in bringing the system to a linear like structure (eg. linearisation or factorisation) of $\mathbf{A}(\hat{\mathbf{x}},\mathbf{u})$ and $\mathbf{H}(\hat{\mathbf{x}},\mathbf{u})$. This difference is illustrated for a scalar output function $z = h(x) \in \mathbb{R}^1$, $x \in \mathbb{R}^1$ in Figure 2.7. The figure shows linearisation around the origin as used in LKF, linearisation around the state estimate as used in EKF and factorisation as used in SDRE, for this scalar function.

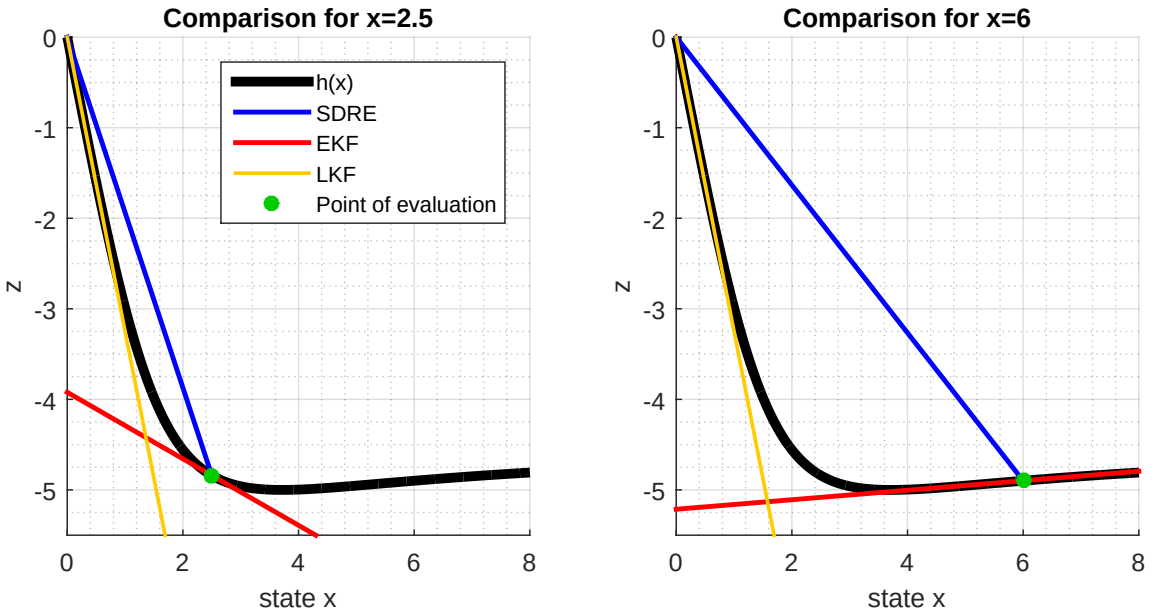


Figure 2.7: Comparison of method for bringing the function to a linear-like structure, illustrated for $x, z \in \mathbb{R}^1$.

In this figure the slopes of the lines labeled LKF, EKF and SDRE algorithms indicate the values for the coefficient H used in (2.37) and (2.38) for $\hat{x} = 2.5$ and $\hat{x} = 6$. As illustrated, LKF linearises around the origin, regardless of current state estimate. However, EKF linearises around the current state estimate, as can clearly be seen by the difference in left and right plot. SDRE uses factorisation such that a proportional function is created to describe the output, which is also dependent on the current state estimate. Further from the origin, the differences between these methods become large, even changing sign for EKF and SDRE. This difference will be used in this section to describe the different behaviour for the different methods. In the remainder of this section the notations $\mathbf{H}_{EKF}$ and $\mathbf{H}_{SDRE}$ shall be used to indicate the values that are used for $\mathbf{H}(\mathbf{x},\mathbf{u})$ for EKF and SDRE respectively.

2.3.1 Innovation Based Correction

The state update equation (2.36) is identical for all algorithms with the exception of the Kalman gain $\mathbf{K}$ from (2.38). Thus the only difference in the state estimates between estimates originates from a different calculation of $\mathbf{H}(\mathbf{x})$ and $\mathbf{P}$. To isolate these effects, a study is proposed for the case where the state that is to be estimated is in fact a parameter. This implies that $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ and thus the problem is only non-linear is the output $\mathbf{h}(\mathbf{x})$. The estimation problem is now only dependent on the selection of matrix $\mathbf{H}(\mathbf{x})$. This is described in the following section. From Figure 2.7 it is observed that $\mathbf{H}_{EKF}(x)$ and $\mathbf{H}_{SDRE}(x)$ can have different signs if the non-linear output map $\mathbf{h}(\mathbf{x})$ is non-convex. This makes these non-convex problems particularly of interest.

Consider the following scalar system with no dynamics and non-linear output map $z = h(x)$.

$$\dot{x} = f(x) = 0, \qquad x \in \mathbb{R}^1 \tag{2.39}$$

$$z = h(x) = -D\sin[C\arctan\{-E(Bx - \arctan(Bx)\}], \qquad z \in \mathbb{R}^1 \tag{2.40}$$

Note that for these cases where $C > 1$ the system is not fully observable following (A.15), since

$$\frac{\partial h(x)}{\partial x} = \frac{B^3CDEx^2\cos[C\arctan\{E(\arctan(Bx) - Bx)\}]}{(B^2x^2+1)(E^2(\arctan(Bx) - Bx)^2 + 1)} \tag{2.41}$$

This derivative becomes zero for $x = x_m$, where x_m is the location of the extremum of (2.40). x_m is the positive valued solution to $C\arctan\{E(\arctan(Bx_m) - Bx_m)\} = \frac{\pi}{2}$, which only has one positive solution for $C > 1$.

This means the system is locally observable on the domains $X_1 = \{x \in \mathbb{R}^1 \mid -x_m < x < x_m\}$, $X_2 = \{x \in \mathbb{R}^1 \mid x < -x_m\}$ and $X_3 = \{x \in \mathbb{R}^1 \mid x > x_m\}$ separately. However, observability is not ensured on the full real domain. Although two points of the system do not satisfy the observability criterion, it is interesting to study the different behaviour of the EKF and SDRE algorithm for the specified system.

The scalar system assumption reduces the multi variable equations (2.36), (2.37) and (2.38) to the scalar equations

$$\dot{\hat{x}} = K(z - h(\hat{x})), \qquad \hat{x} \in \mathbb{R}^1 \tag{2.42}$$

$$\dot{P} = G - P^2H^2V^{-1} \qquad P \in \mathbb{R}^1 \tag{2.43}$$

$$K = PHV^{-1} \qquad K \in \mathbb{R}^1 \tag{2.44}$$

Stationary points can be identified for $z = h(\hat{x})$ and $H(x) = 0$. Note that since P represents the error covariance matrix it is positive definite and therefore can never become zero in the presence of process noise. Similarly, V is finite, thus V^{-1} can also never become zero. The identified stationary points can be seen to be the only possible points to which the estimate can converge as $t \to \infty$. Note also that the stability of these equilibria depends on the true output z.

Perfect Model

First, the case were the output is perfectly modelled is considered. The proposed system is initialised at various true states and initial estimates. Four classifications are proposed to qualitatively describe the behaviour over infinite time. Firstly, convergence to the correct state $\hat{x} \to x$. The second and third classification both indicate convergence, but to an incorrect estimate. The difference between the two being if the initial error has decreased or increased. The final classification is exploding of the estimate, implying that $\hat{x} \to \infty \vee -\infty$ for $t \to \infty$. The selected output map is shown in Figure 2.8, with extrema $x_m \approx 3.714$. The convergence properties for given initial conditions are then given in Figure 2.9.

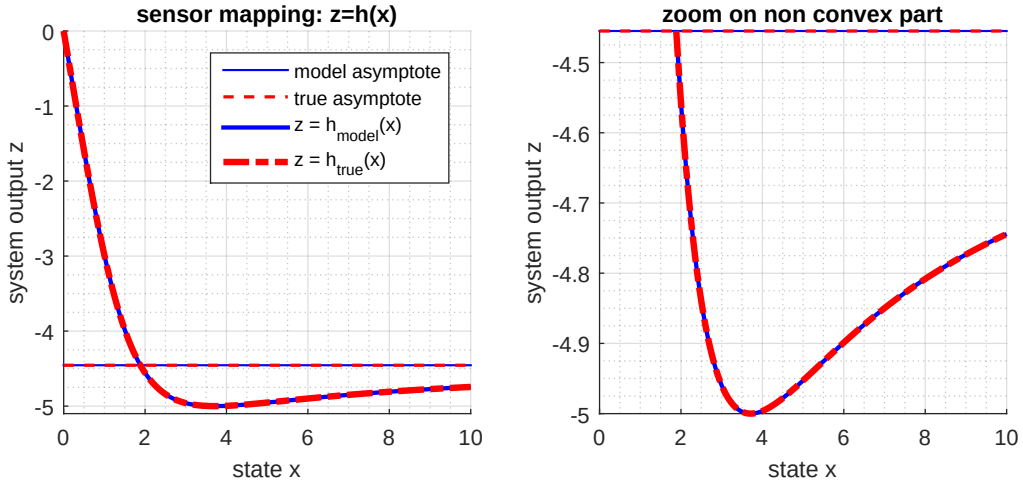


Figure 2.8: Output $h(x)$. The outputmap is assumed to be known such that $D_{mod} = D_{true} = 5, C_{mod} = C_{true} = 1.3, B_{mod} = B_{true} = 0.5, E_{mod} = E_{true} = -1$, an extremum is located at $x_m \approx 3.714$

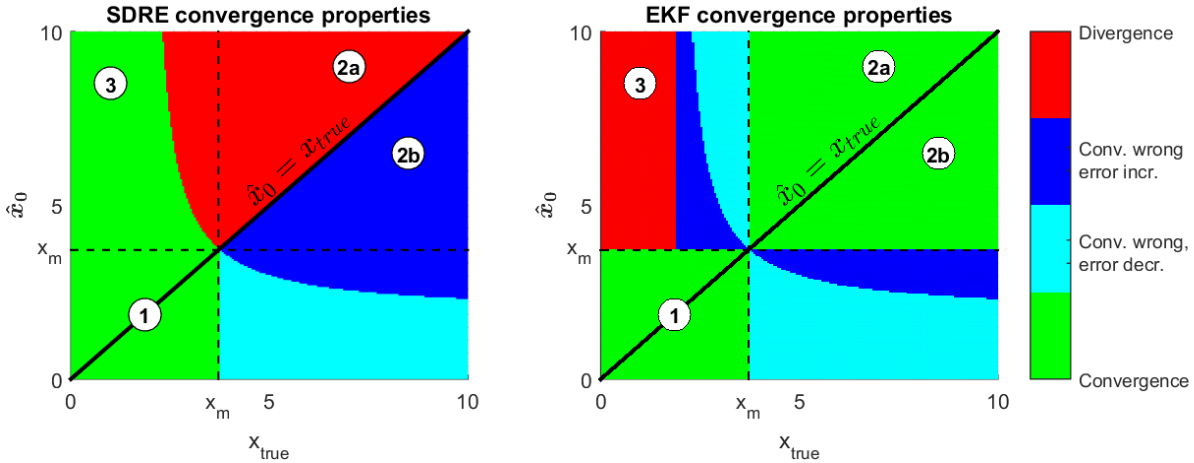


Figure 2.9: Comparison of filter algorithms $\hat{x}_0$ and true states x_{true}, for perfect model: $D_{mod} = D_{true} = 5, C_{mod} = C_{true} = 1.3, B_{mod} = B_{true} = 0.5, E_{mod} = E_{true} = -1$

It can be recognised that even in the static case, convergence is very different between the SDRE and the EKF. This difference can be contributed to the change in sign for H_{EKF} when $\hat{x} > x_m$, with x_m being the location of the extremum in $h(x)$. This is supported by the fact that the classifications in Figure 2.9 for $\hat{x}_0 < x_m$ are identical, since in this section $\operatorname{sign} H_{EKF} = \operatorname{sign} H_{SDRE}$.

As described previously, H_{EKF} and H_{SDRE} dictate the direction of convergence. More precisely, it indicates in which direction of the estimate the true state is expected to be for a given difference in measured and predicted output. Consider again (2.42) and substitute the expression for K and z to get (2.45). Also recall that PV^{-1} is positive definite.

$$\dot{\hat{x}} = PV^{-1}H(h(x) - h(\hat{x})) \tag{2.45}$$

Several examples shall be discussed regarding various cases as indicated in Figure 2.9.

Case 1 First, consider the case where $\hat{x}_0, x_{true} < x_m$, corresponding to the lower left corners of Figure 2.9. Since $\hat{x}_0 < x_m$, $H_{EKF}, H_{SDRE} < 0$. The combination $h(x) < h(\hat{x})$ and $H < 0$ yields an increase in $\hat{x}$, whereas the combination $h(x) > h(\hat{x})$ and $H < 0$ yields a decrease in $\hat{x}$

(2.45). This increase or decrease will continue until either $h(x_{true}) = h(\hat{x})$ or $H = 0$. On the specified bounds of $x_{true} < x_m$ the first criteria will be encountered first. The result is that both EKF and SDRE will converge to the right value.

Case 2a Consider $\hat{x}_0, x_{true} > x_m$, corresponding to the upper right corners of Figure 2.9. For these conditions holds $H_{EKF} > 0$, while $H_{SDRE} < 0$. Now consider first $\hat{x}_0 > x_{true}$, which results in $h(x_{true}) < h(\hat{x})$. Therefore, $\dot{\hat{x}}_{SDRE} > 0$ and $\dot{\hat{x}}_{EKF} < 0$. This will continue to either $h(x_{true}) = h(\hat{x})$ or $H = 0$. It should be noted however that neither will be the case for the SDRE, since the output error only increases. This results in $\hat{x}_{SDRE} \to \infty$ as $t \to \infty$. The EKF will reduce the estimate and converges to the right value. This is seen in the top right corners above the diagonals.

Case 2b Again $\hat{x}_0, x_{true} > x_m$. However, now consider $\hat{x}_0 < x_{true}$ and thus $h(x_{true}) > h(\hat{x})$. Again, EKF will update the estimate in the correct direction and converge to $h(x_{true}) = h(\hat{x})$, which for the specified bounds corresponds to $\hat{x} > x_{true}$. In contrast, SDRE will decrease the state estimate, until $h(x_{true}) = h(\hat{x})$. For this case there is an equilibria for the SDRE to converge to, this is however, not the correct estimate. This is seen in the top right corners below the diagonals.

Case 3 It should be noted that the EKF is also able to explode. This happens when $\hat{x}_0 > x_m$ and $h(x_{true}) > \lim_{x\to\infty} h(x)$. However, the SDRE will still be able to converge as the factorisation drives the estimate towards the origin whenever $h(x_{true}) < h(\hat{x})$. This is seen in the upper left area of Figure 2.9.

In summary, the above shows that even for static parameter estimation a significant difference between SDRE and EKF exists. This can be attributed to the shape of the output map $h(x)$, which results in two points not satisfying the observability criterion. It also shows the effect of the initial condition on convergence properties of the algorithms. Although both algorithms show strengths and weaknesses in different regions, it can be argued that the EKF is better suited for this example. Since normally the initial condition should be close to the true state. This corresponds to the bottom left to top right diagonal in Figure 2.9.

Imperfect Convex Model

Although the previous section gives a clear view of the functionality of the algorithm, it could be argued that in reality the output-mapping $z = h(x)$ is not known exactly, hence $h_{true}(x) \neq h_{model}(x)$. The previous section demonstrates that the non-convex model results in issues for some combinations of initial estimate and true state. In an attempt to prevent these issues a convex model is suggested. For that reason the map is fitted using a convex assumption. Linear behaviour for small states is also matched. Finally, parameter E is tuned to match behaviour around the peak better. An example of an output-mapping for such a case is shown in Figure 2.10. It should be noted that true convergence can only occur at intersections for which $h_{true}(x) = h_{model}(x)$. Therefore, practical convergence is defined as $\hat{x} = x_{true} \pm \epsilon$, with ϵ the desired tolerance of estimation.

Figure 2.11 shows that for a convex model, there is no longer a distinction between the EKF and SDRE algorithms. This corresponds well with the expectations as H_{EKF} and H_{SDRE} now can no longer have different signs.

Imperfect Non-Convex Model

Next, the case is considered where the output is modelled with a non-convex model as is the case in Figure 2.9, but limitations in modelling limit the accuracy of the model. This is illustrated in Figure 2.12.

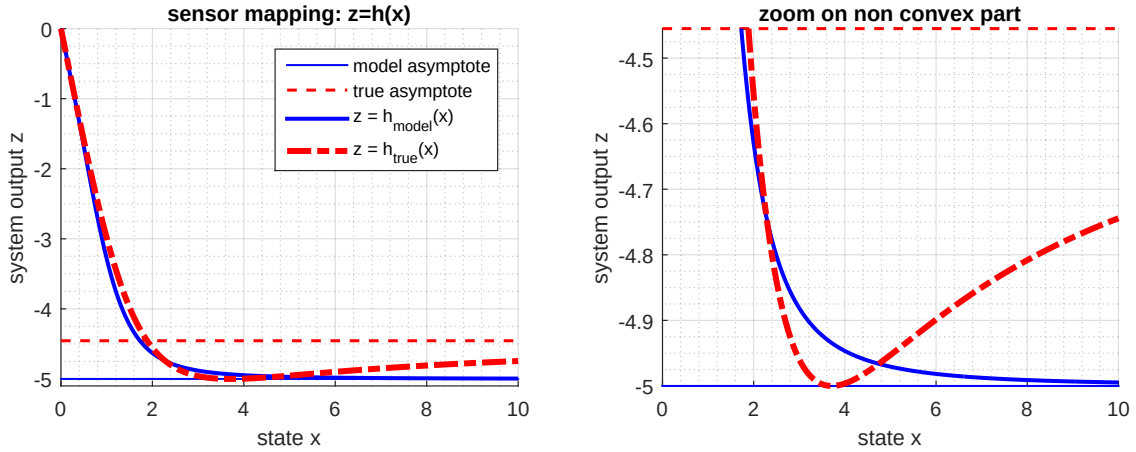


Figure 2.10: Outputmap $h(x)$. The outputmap approximation. $D_{mod} = D_{true} = 5, C_{mod} = 1, C_{true} = 1.3B_{true} = 0.5, B_{mod} = D_{true}B_{true}C_{true}/(D_{mod}C_{mod}), E_{mod} = -3, E_{true} = -1$, an extremum is located at $x_m \approx 3.714$

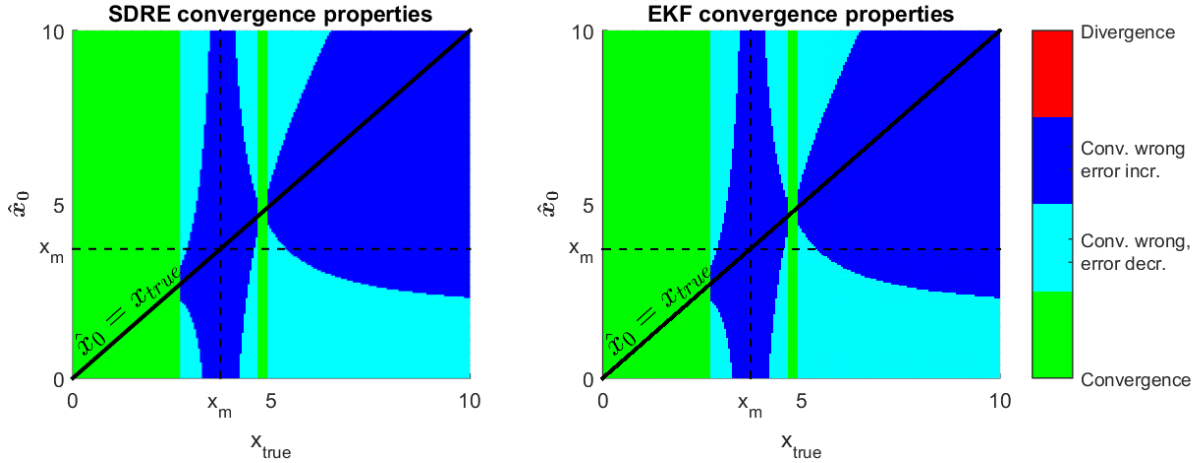


Figure 2.11: Comparison of filter algorithms $\hat{x}_0$ and true states x_{true}, for imperfect model: $D_{mod} = D_{true} = 5, C_{mod} = 1, C_{true} = 1.3B_{true} = 0.5, B_{mod} = D_{true}B_{true}C_{true}/(D_{mod}C_{mod}), E_{mod} = -3, E_{true} = -1$, tolerance $\epsilon = 0.5$

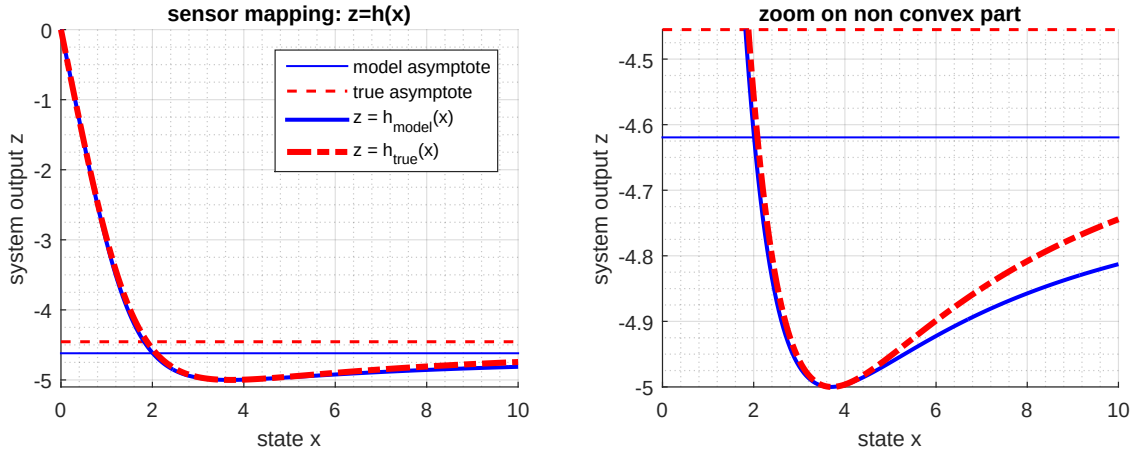


Figure 2.12: Output $h(x)$. The outputmap approximation. $D_{mod} = D_{true} = 5, C_{mod} = 1.25, C_{true} = 1.3B_{true} = 0.5, B_{mod} = D_{true}B_{true}C_{true}/(D_{mod}C_{mod}), E_{mod} = -1.4, E_{true} = -1$, an extremum is located at $x_m \approx 3.714$

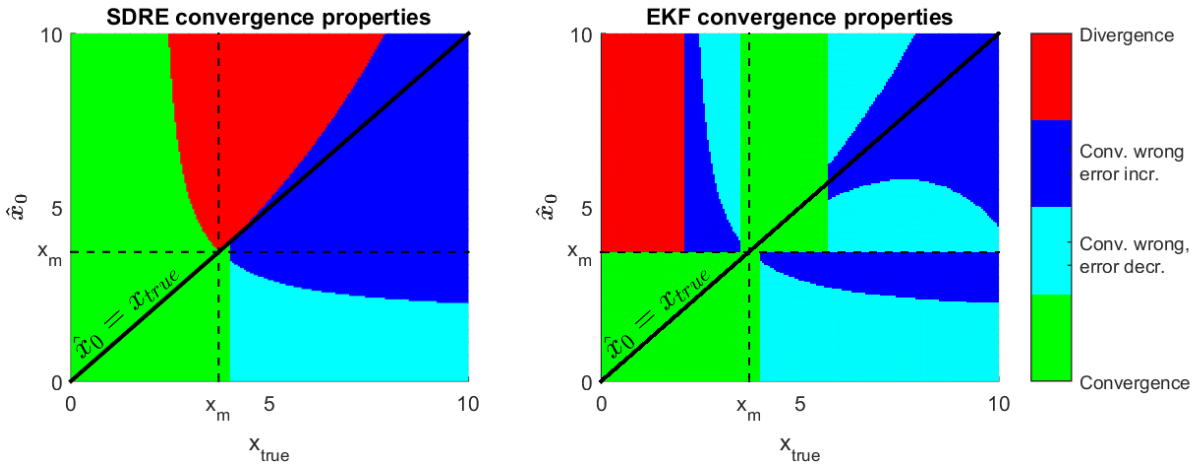


Figure 2.13: Comparison of filter algorithms $\hat{x}_0$ and true states x_{true}, for imperfect model: $D_{mod} = D_{true} = 5, C_{mod} = 1.25, C_{true} = 1.3 B_{true} = 0.5, B_{mod} = D_{true}B_{true}C_{true}/(D_{mod}C_{mod}), E_{mod} = -1.4, E_{true} = -1$, tolerance $\epsilon = 0.5$

The resulting Figure 2.13, now shows that for larger true states the EKF is no longer able to converge to the correct value. This is to be expected as Figure 2.12 clearly shows larger deviations from the true outputs for larger states.

2.3.2 Error Covariance

So far only infinite time results have been considered. However, the transient behaviour is also clearly different between EKF and SDRE. This is the results of different matrices being used in SDRE and EKF for $\mathbf{A}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$, as can be recognised in (2.37).

Algorithms based on Kalman filtering like EKF and SDRE all use a quantity to define the uncertainty in state estimates. For linear systems this is fully captured by the error covariance matrix $\mathbf{P}$. However, it should be noted that for non-linear systems this error covariance is only an approximation of the uncertainty. This is illustrated in appendix A.3 and [26].

It can be shown that even in the static case, the transient behaviour is clearly different between the SDRE and EKF algorithms. To illustrate this, various initial conditions are tested for the convex model from Figure 2.10. This model results in identical steady-state solutions for EKF and SDRE and is selected to highlight the differences in transient behaviour.

The initial conditions for the EKF and SDRE algorithm are selected identically. Hence, $P_{0_{SDRE}} = P_{0_{EKF}}$, $\hat{x}_{0_{SDRE}} = \hat{x}_{0_{EKF}}$. Information regarding the process and measurement noise is also taken identically, $G_{SDRE} = G_{EKF}$, $V_{SDRE} = V_{EKF}$. The resulting time response of the estimate is shown in Figure 2.14.

A few interesting observations can be made. Obviously, when considering (2.43), for a fixed G and V the slope described by $H(x)$ dominates the value to which the error covariance P is converging. Note that in this example no dynamics are considered and hence $A(x) = 0$ for both SDRE and EKF. In the linear case, the slope of $H(x)$ can be considered as the sensitivity of the output to a difference in state. A steeper slope (and hence bigger magnitude of $H(x)$) indicates a more sensitive output and more accurate resulting estimate. This results in turn in lower error covariance, which can also be recognised here. The SDRE factorisation has a much steeper slope than the Jacobian of the EKF. Therefore, the error covariance reduces much faster. In contrast, the steeper slope also results in a less aggressive correction. In this example, these combined effects result in the SDRE converging much slower than the EKF algorithm.

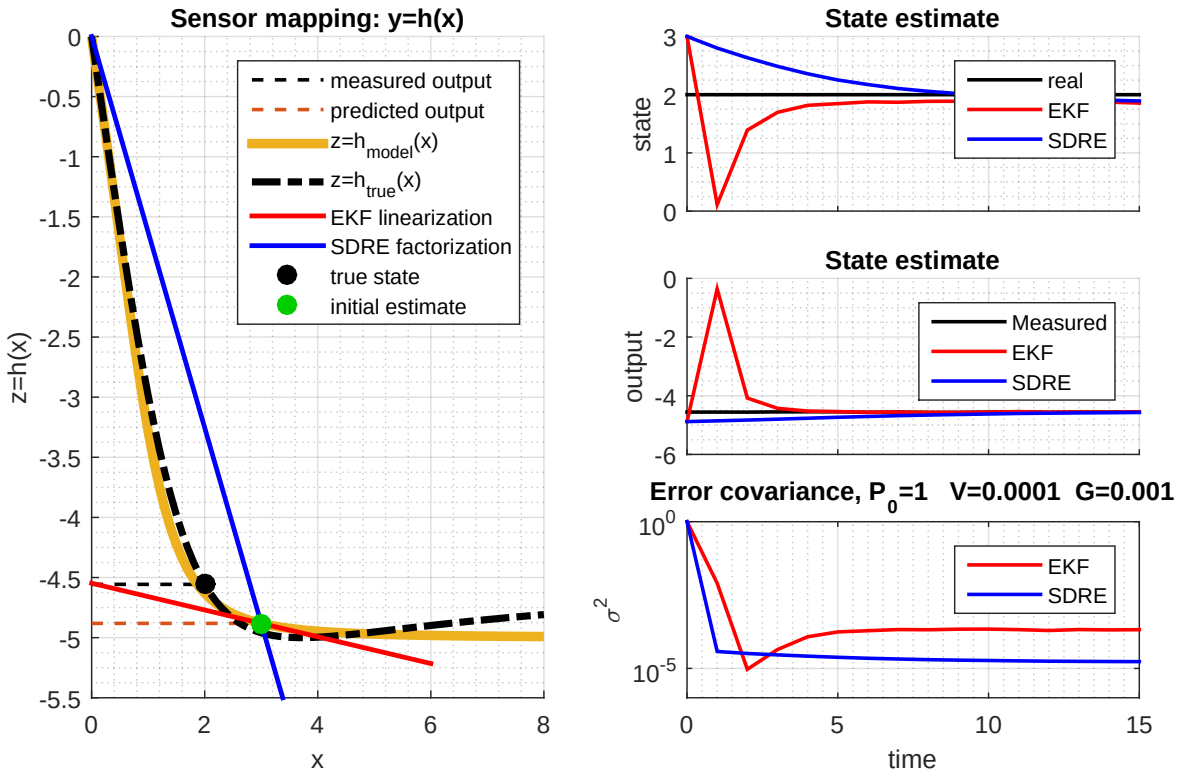


Figure 2.14: Comparison time response of filter algorithms $\hat{x}_0 = 3$ and true state $x_{true} = 2$, for imperfect model: $D_{mod} = D_{true} = 5, C_{mod} = 1.25, C_{true} = 1.3 B_{true} = 0.5, B_{mod} = D_{true} B_{true} C_{true} / (D_{mod} C_{mod}), E_{mod} = -1.4, E_{true} = -1$

2.4 Summary

This chapter has presented the theory regarding state estimation. Several basic principles are explained. The algorithms for the Linear Kalman Filter, Extended Kalman Filter and State Dependent Riccati Equation Estimator are presented.

Some of the results presented literature that compares the different estimators have been reproduced and extended. This illustrated the importance of selecting the initial condition of the error covariance matrix. Also the difference between usage of the ARE and DRE are illustrated for the pendulum problem. Additionally, some of the conclusions presented in the literature are put into context.

Some of the fundamental differences between the SDRE and EKF are also discussed and presented using an illustrative scalar system description. A visual interpretation of the linearisation and factorisation used in respectively the SDRE and EKF are given for a scalar function. To isolate the functionality the estimator from the modelled dynamics, parameter estimation with absence of dynamics is studied. The various steps of the estimator algorithms are isolated and compared for this parameter estimation. Clear differences could be observed between the SDRE and EKF estimator.

It should be noted that no direct conclusions can be drawn about the performance of either of the algorithms at this point. The presented simplifications that are assumed in each example will in reality hardly ever occur. Nevertheless, the presented comparison does give insight in the functionality of the algorithms.

Chapter 3

Observer model

Estimating the state of any system requires a model of the dynamics and measurements as illustrated in Figure 2.1. This chapter describes the model that is used in the Vehicle State Estimator. First the dynamics and equations related to the chassis model are described. Secondly, the modelling of the tyre characteristics are discussed. It is concluded with a parameter identification and model validation part.

3.1 Vehicle model

Depending on the available information of the vehicle and the objective for the model, the vehicle model can vary greatly in complexity. This ranges from simple point-mass models to highly detailed multi-body models. The following section describes the used model and briefly discusses possible extensions of the model. Finally, a motivation for using the selected model is given.

3.1.1 Planar Vehicle Model: Two-track Model

Planar vehicle models only consider in-plane dynamics. Vertical dynamics are neglected and the out-of-plane normal forces are replaced by their steady-state representations. A rigid mass is actuated by a number of forces, which represent the vehicle's tyres. When relevant, additional body forces can be added to model aerodynamic loads. In the scope of this project only four tyres are considered, but the general planar vehicle model can easily be extended to include additional tyres when analysing for instance trucks.

A common planar vehicle model is the two-track model. As the name suggests this model consists of two 'tracks' along the length of the vehicle. Having two tracks implies that lateral load transfer can be taken into account. Additionally, since the wheels are positioned at a lateral distance with respect to the center of gravity, the longitudinal forces introduced at these wheels will result in a yaw moment. Naturally, the same is true for lateral forces since a longitudinal distance of the wheels with respect to the center of gravity is also present. A graphical representation of the two-track model is given in Figure 3.1.

Equations of Motion

The two track model has three degrees of freedom (DOF's) being the in plane velocities. These velocities are the yaw rate r, which is the rotational velocity around the z-axis, longitudinal velocity V_x and lateral or side slip velocity V_y. The equations of motion corresponding to these states are given in (3.1).

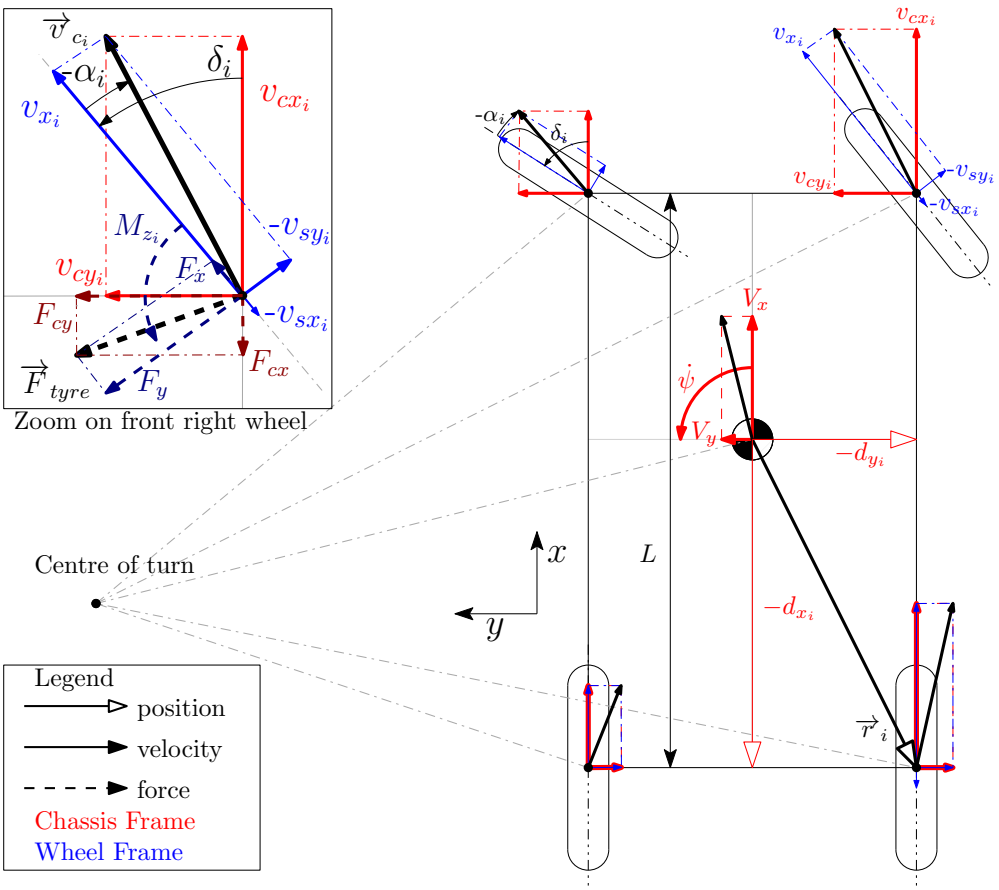


Figure 3.1: Two track model representation. Velocities are given with solid lines, forces are given by dashed lines in the zoomed section. Wheel positions wrt. the COG are indicated with arrows with open tips.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{r} \\ \dot{V}_y \\ \dot{V}_x \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} \frac{M_{z_i} + d_{y_i} F_{cx_i} + d_{x_i} F_{cy_i}}{I_{zz}} \\ \sum_{i=1}^{4} \frac{F_{cy_i}}{m} \\ \sum_{i=1}^{4} \frac{F_{cx_i}}{m} - \frac{F_{\text{drag}}}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ -rV_x \\ rV_y \end{bmatrix} \tag{3.1}$$

Note that the force components in these equations are defined in the chassis frame. Typically, the forces generated by the tyre are defined by the tyre model in the wheel plane. These forces are also dependent on velocity and slip definitions in this same plane. To evaluate tyre forces it is therefore required to know the velocities in this wheel plane. These velocities can be found by finding the speeds in the chassis frame coordinates, and rotating about the wheel's angle δ_i.

$$\begin{bmatrix} v_{cx_i} \\ v_{cy_i} \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \end{bmatrix} + r \begin{bmatrix} -d_{y_i} \\ d_{x_i} \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} v_{x_i} \\ v_{sy_i} \end{bmatrix} = \begin{bmatrix} \cos\delta_i & -\sin\delta_i \\ \sin\delta_i & \cos\delta_i \end{bmatrix} \begin{bmatrix} v_{cx_i} \\ v_{cy_i} \end{bmatrix} \tag{3.3}$$

Here, v_{cx,y_i} and d_{x,y_i} are components of respectively the velocity and the position vector in the chassis frame for wheel i (see again Figure 3.1 for clarification). Equation (3.3) provides the velocity components in the wheel plane, with v_{sy_i} the lateral slip velocity. The longitudinal slip velocity can be calculated by taken into account the rotational velocity of the wheel ω_i in

combination with the effective rolling radius R_{e_i}.

$$v_{sx_i} = v_{x_i} - R_{e_i}\omega_i \tag{3.4}$$

Aside from velocities, the generated tyre force strongly depends on the normal load acting on the wheel. This normal load has a static term that depends on the position of the wheels with respect to the center of gravity. Additionally, various terms due to load transfer are present. Under the assumption that the vehicle is symmetric we find

$$F_{\text{nom}_{z_i}} = \frac{mg(L - |d_{x_i}|)}{2L} \tag{3.5}$$

$$\Delta F_{z_{a_x}} = \frac{-a_x m h_{\text{cog}}}{2L} \tag{3.6}$$

$$\Delta F_{z_{a_y}} = \frac{-a_y m h_{\text{cog}}}{2L_{tw}} \tag{3.7}$$

$$\Delta F_{z_{\text{drag}}} = \frac{-F_{\text{drag}} h_{\text{cop}}}{2L} \tag{3.8}$$

$$F_{z_i} = F_{\text{nom}_{z_i}} + \text{sign}\,(d_{x_i})\Delta F_{z_{\text{drag}}} + \text{sign}\,(d_{x_i})\Delta F_{z_{a_x}} + \text{sign}\,(d_{y_i})\Delta F_{z_{a_y}} \frac{k_{\phi_i}}{k_{\phi_{\text{tot}}}} \tag{3.9}$$

Here the longitudinal load transfer is assumed to be equally distributed over the left and right track, since the pitch stiffnesses contributions of left and right track are typically identical. The lateral load transfer is influenced by the roll stiffness coefficient. Although body roll is not modelled in the planar vehicle model, the load transfer due to lateral acceleration is influenced by the ratio of roll stiffness of that axle with respect to that of the entire vehicle. Note that these equations are only valid until the point that one of the normal loads becomes negative, as a wheel would then lift off from the ground. From that point onward, any additional lateral load transfer will be subjected to the axle that still has a load on both wheels. Note that in such an event the lifted wheel does not generate a tyre force. Therefore, only the remaining tyres influence the dynamics. Some loss of accuracy can be expected due to for example larger camber angles, but the equations of motion still remain valid.

The tyre forces generated by each tyre can now be calculated using any tyre model. This will be elaborated on in Section 3.2. For now it is sufficient to know that the reaction forces of the tyre due to wheelslip are a function of the speed components at the wheel and the normal load. The tyre forces and aerodynamic drag force [27] are given by

$$\mathbf{F}_{\text{tyre}_i} = \begin{bmatrix} F_{x_i} \\ F_{y_i} \\ M_{z_i} \end{bmatrix} = \begin{bmatrix} F_{x_i}(v_{x_i}, v_{sy_i}, v_{sx_i}, F_{z_i}) \\ F_{y_i}(v_{x_i}, v_{sy_i}, v_{sx_i}, F_{z_i}) \\ M_{z_i}(v_{x_i}, v_{sy_i}, v_{sx_i}, F_{z_i}) \end{bmatrix} \tag{3.10}$$

$$F_{\text{drag}} = \frac{1}{2} C_{d_{\text{eff}}} V_x^2, \qquad C_{d_{\text{eff}}} = A_{\text{drag}} C_d \rho_{\text{air}} \tag{3.11}$$

Where A_{drag} denotes the drag area, C_d the drag coefficient of the car and ρ_{air} the air density. The tyre forces are defined in the wheel plane and can be transformed to the chassis frame by

$$\begin{bmatrix} F_{cx_i} \\ F_{cy_i} \\ M_{z_i} \end{bmatrix} = \begin{bmatrix} \cos\delta_i & \sin\delta_i & 0 \\ -\sin\delta_i & \cos\delta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{x_i} \\ F_{y_i} \\ M_{z_i} \end{bmatrix} \tag{3.12}$$

Model Inputs

Multiple inputs can be selected for the model dependent on the level of information that is available. They can be roughly grouped in kinematic inputs and force based inputs. When information regarding the moment applied to the wheels is known the model can be extended by 4 DOF's, to include the four individual wheelspeeds. This requires however, that the torques applied to the wheels are known, which in turn requires that the brake pressures in the individual callipers are known and that the drive-line torque is also modelled. This is however not always the case, as often only brake pressure in the main piston is known and individual brake pressures are controlled by the ABS unit. The alternative is using the wheelspeed as an input to the model, which has as a disadvantage that this results in an input signal with more noise.

For the steering angle of each wheel, typically the steering wheel angle is used. The steering angle of each individual wheel can then easily be calculated using the kinematic relations for the vehicle. However, it is important to consider that the suspension system is not infinitely stiff. Hence, the steering angle of each wheel will be influenced by the in-plane forces generated by that tyre. This can be modelled by introducing an additional steering angle as function of lateral force and self aligning moment

$$\delta_i = \delta_{\text{kin}_i}(\delta_{sw}) + \delta_{\text{com}_i}(F_{cy_i}, M_{z_i}) \tag{3.13}$$

$$\delta_i = \delta_{\text{kin}_i}(\delta_{sw}) + K_{\delta M_z} M_{zkp_i} \tag{3.14}$$

The kinematic relation between steering wheel angle δ_{sw} and kinematic wheel angle δ_{kin_i} is characterised by the steering ratio and steering system geometry. Typically, the inner wheel should have a larger steering angle, due to the inside track running on a smaller radius. This is achieved by Ackerman geometry [27], as illustrated in Figure 3.2a. A case is illustrated in which the angle γ_{ack} aligns the pivot point of the steering rod with the center of the rear axle. This is referred to as 100% Ackerman. Angle γ_{ack} can be used to derive the kinematic relation $\delta_{\text{kin}_i}(\delta_{sw})$. The following equations can be derived for the front left and front right steering angles. This relation is illustrated in Figure 3.2b

$$\gamma_{\text{ack}} = \arctan(r_{\text{ack}} \frac{L_{tw}}{2L}), \qquad \gamma_{\text{ack}_l} = \gamma_{\text{ack}}, \qquad \gamma_{\text{ack}_r} = -\gamma_{\text{ack}} \tag{3.15}$$

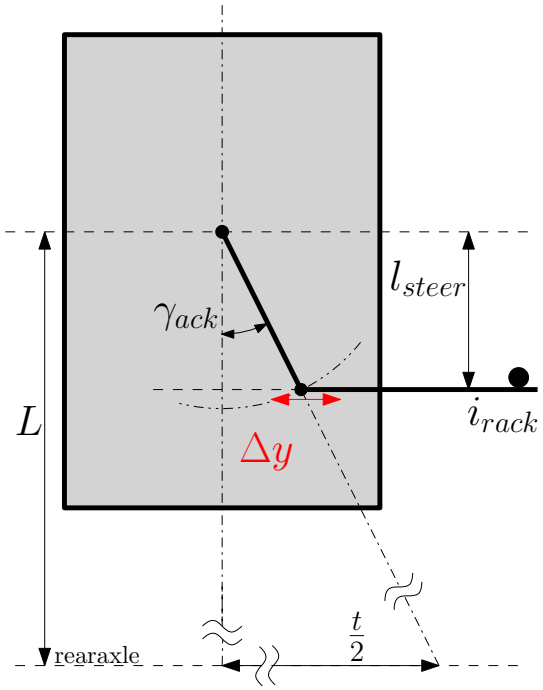
$$\delta_l = \arcsin\left(\frac{\delta_{sw}}{i_{\text{steer}}}\cos(\gamma_{\text{ack}_l}) + \sin(\gamma_{\text{ack}_l})\right) - \gamma_{\text{ack}_l} \tag{3.16}$$

$$\delta_r = \arcsin\left(\frac{\delta_{sw}}{i_{\text{steer}}}\cos(\gamma_{\text{ack}_r}) + \sin(\gamma_{\text{ack}_r})\right) - \gamma_{\text{ack}_r} \tag{3.17}$$

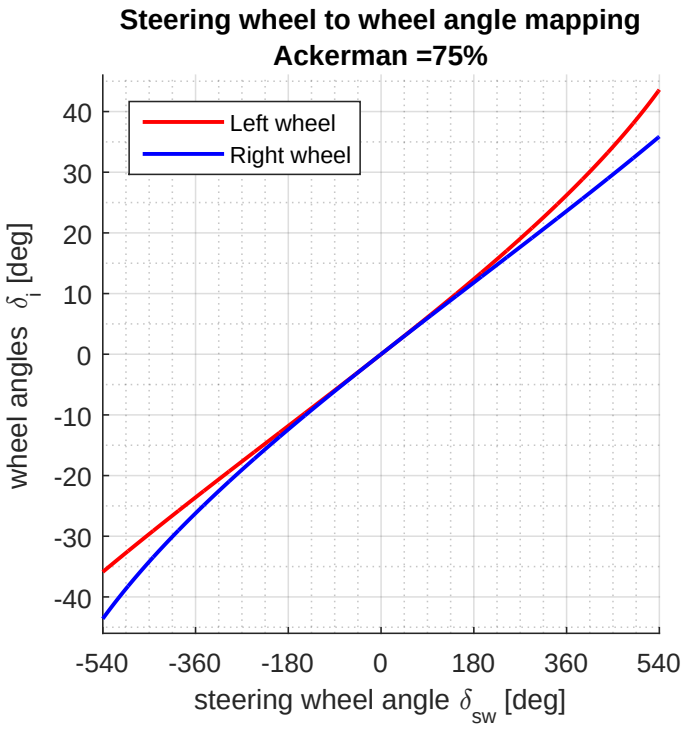
These kinematic relations result in Figure 3.2b, where $i_{\text{steer}} = 15.648$ and $r_{\text{ack}} = 0.75$, $L_{tw} = 1.519$ $L = 2.7$. It can be seen that when the steering wheel is turned left (postive δ_{sw}) the left wheel attains a bigger angle.

3.1.2 Roll-Axis Model

The roll-axis model is an extension to the planar vehicle model in the sense that it adds a roll angle DOF. This allows for including roll (and if desired pitch) dynamics in the equations of motion. The added complexity of this model has a few advantages. Firstly, since roll is now an included state, the sensor reading from the body mounted IMU (Inertial Measurement Unit) can be described more accurately. Additionally, a better estimate of road bank angle can be obtained. Also, the normal forces subjected to the tyres can be modelled directly as function of roll angle and rate, and pitch angle and rate by using stiffness and damping properties of the suspension. This removes the algebraic loop that is present in the planar model. A graphical representation of the roll-axis model can be found in Figure 3.3.



(a) Steering geometry, pivot point of steering rod on upright results in angle γ_{ack}, which can be used to derive the kinematic relation between δ_{sw} and δ_i



(b) Mapping of steering wheel angle to wheel angle. Note 75% ackerman is adopted on 2012 toyota prius.

Figure 3.2: Ackerman Steering Geometry

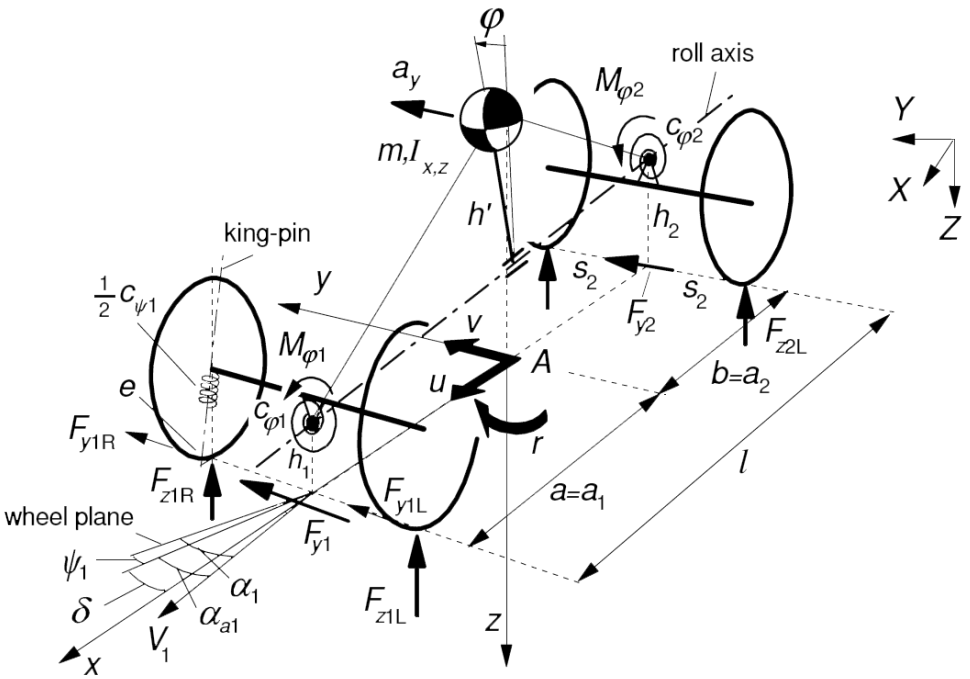


Figure 3.3: Representation Roll-axis model [28], the suspended mass is able to roll around the roll-axis defined by suspension geometry. Normal loads are calculated by combining roll angle and rate with stiffness and damping properties of the suspension.

3.1.3 Model Motivation

Within the scope of this project, emphasis is placed on the yaw dynamics under the effects of combined braking and steering. This is accurately captured by the two track model, which is capable of describing the yaw dynamics introduced by asymmetric braking inputs. Figure 3.4 and Figure 3.5 show a comparison between the behaviour of the bicycle, two track and roll axis model, for parameters matching a Toyota Prius with Magic Formula [28] tyre model.

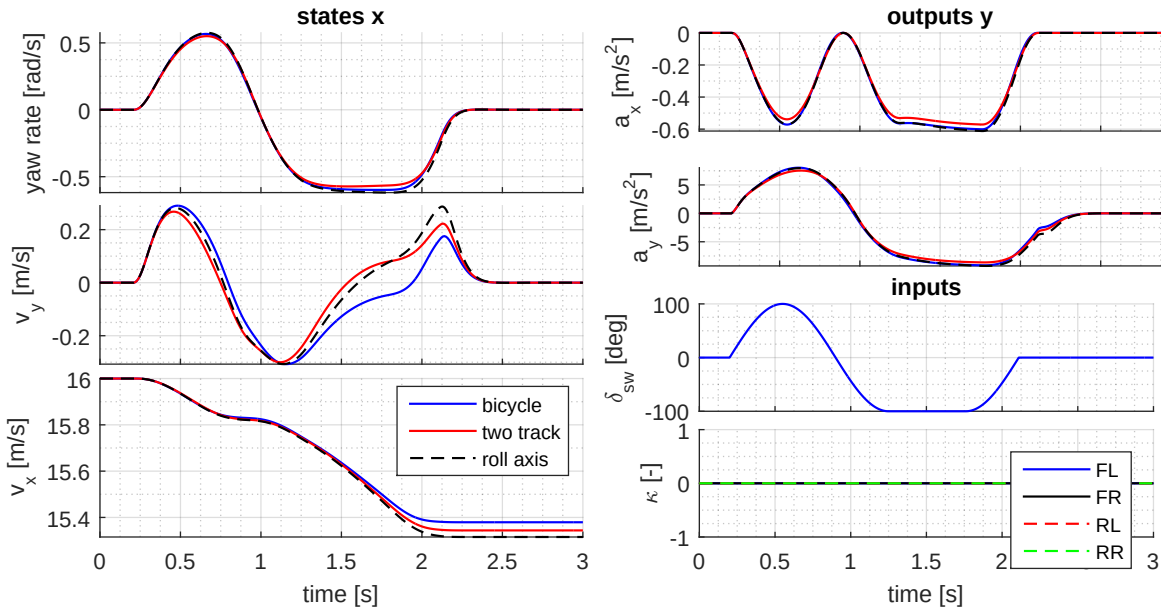


Figure 3.4: steering manoeuvre

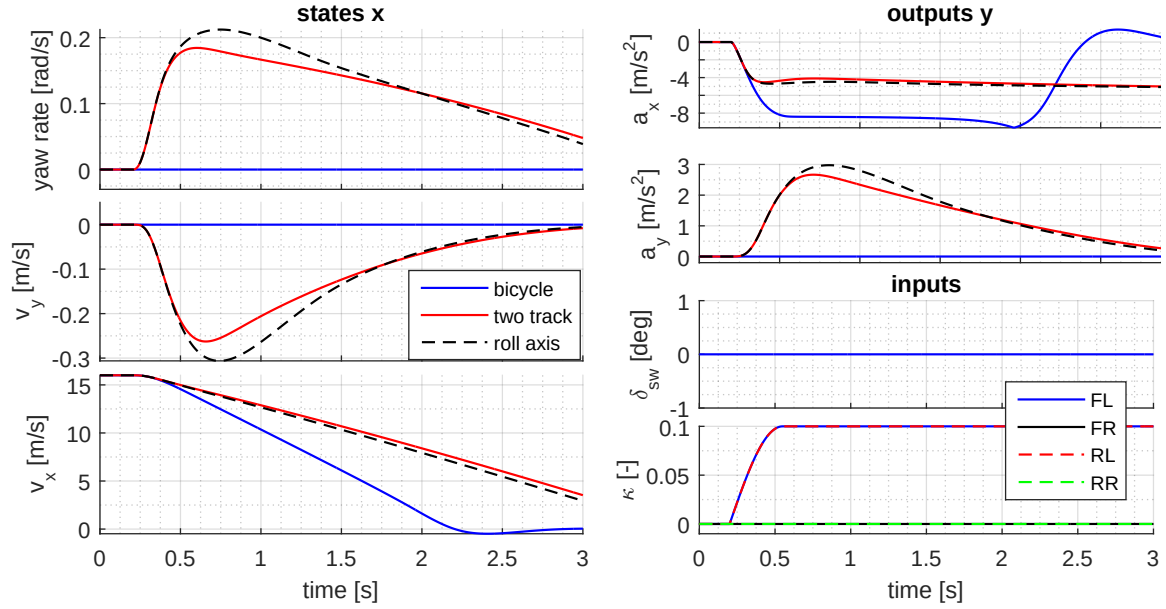


Figure 3.5: asymmetric braking

The sine with dwell steering manoeuvre shows very similar behaviour between all three models. However, when asymmetric braking is applied, as can be the case for collision avoidance actuation, the bicycle model is clearly not capable of capturing the yaw dynamics. The roll axis model show some discrepancies with the two track model. However, the two track model describes all the relevant dynamics without the added complexity of the roll-axis model. Therefore the two track model is used for implementation in the vehicle state estimator.

3.2 Tyre Modelling

As mentioned in section 3.1, a proper model of the tyre behaviour is required for the vehicle model. The tyres play a crucial role in the dynamics of the vehicle, since all forces (with the exception of aerodynamic forces) acting on the vehicle are generated by the tyres. Due to the large impact of tyres on the vehicle dynamics, tyres have been studied extensively in literature. Many different models exist to describe the forces generated by the tyres, varying from empirical curve fitting to physical models that attempt to capture the physics involved. In the context of vehicle state estimation, the interest in the tyre model is twofold. Firstly, an accurate model enables to accurately capture the dynamics. This improves the vehicle's response under the influence of the model inputs. Secondly, observer algorithms as the EKF require a Jacobian to be calculated, which is also influenced by the choice in tyre model.

This section will briefly discuss some tyre models under consideration for usage in the vehicle state estimator. A motivation for the choice on the used tyre model will then be given, together with the procedure for fitting the parameters.

3.2.1 Brush Model

The brush model is an insightful physical model as presented in [28]. The brush model utilises the pressure distribution in the contact patch to describe the maximum deformation of elastic *bristle* elements. Typically, a contact length based parabolic pressure distribution is assumed, in which the contact length is an empirical function of vertical load. The pure slip case is illustrated very clearly in Figure 3.6.

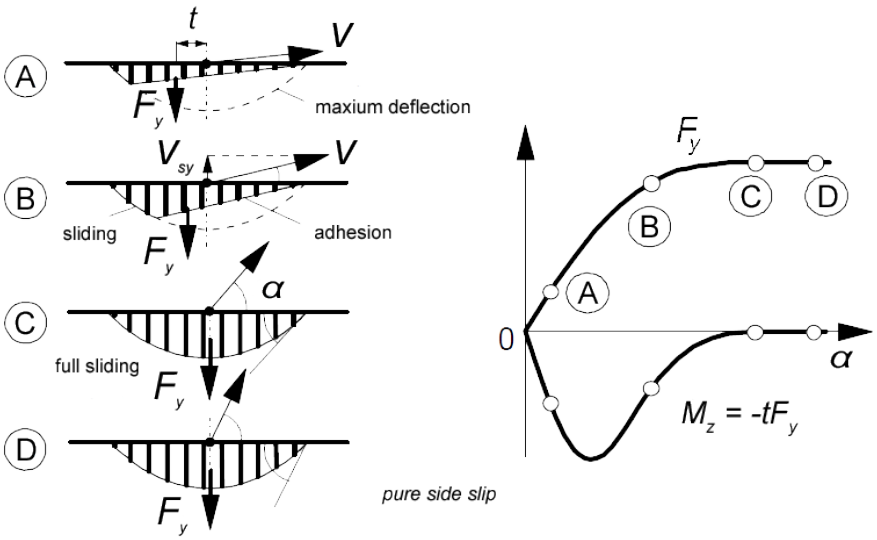


Figure 3.6: Illustrative explanation of brush model. [28] maximum bristle deformation is function of the parabolic pressure distribution. Integration of the bristle deformation length results in lateral tyre force.

Bristles entering the contact patch initially have no deformation. Once a bristle element has entered the contact patch it starts deforming proportionally to the distance to the leading edge, with a gradient equal to the slip angle. This continues until the bristle reaches the maximum deformation described by the parabolic pressure distribution. The force generated by the tyre is then the sum of all bristle elements. An extension to combined slip conditions is fairly straightforward. A disadvantage of the model is that it uses logic to define slip conditions (eg. full sliding), making the calculation of the derivative unattractive.

3.2.2 Exponential Tyre Model

A tyre model that shows very similar behaviour as the brush model is the exponential tyre model. This model is currently used in the current TNO VSE [29]. A function showing a similar shape as the brush model is used. An advantage of this model is that with only four parameters, both the linear low slip behaviour as well as tyre saturation are captured. Additionally, differentiation of a natural exponential function is very straightforward. Note also the addition of a small value ϵ in (3.18), which prevents numerical issues when the total slip approaches zero.

$$S = \sqrt{\alpha^2 + \kappa^2 + \epsilon} \tag{3.18}$$

$$F_x = \mu_x F_z \frac{\kappa}{S}(1 - \exp^{-K_x|S|}) \operatorname{sign} \kappa \tag{3.19}$$

$$F_y = \mu_y F_z \frac{\alpha}{S}(1 - \exp^{-K_y|S|}) \operatorname{sign} \alpha \tag{3.20}$$

Vehicle state estimation in case of collision avoidance implies limited handling operating conditions. It could be of interest to capture the true tyre potential better. An extension to the before mentioned exponential tyre model is presented in [30]. It involves a second exponential term, allowing it to describe a peak force and high slip force drop off. This results in a tyre model of the following shape in pure slip conditions.

$$F_x = \mu_x F_z \frac{\kappa}{S}\left(E_{ex}|S|e^{-b_{ex}|S|} + C_{ex}(1 - e^{-b_{ex}|S|})\right) \operatorname{sign}(\kappa) \tag{3.21}$$

$$F_y = \mu_y F_z \frac{\alpha}{S}\left(E_{ey}|S|e^{-b_{ey}|S|} + C_{ey}(1 - e^{-b_{ey}|S|})\right) \operatorname{sign}(\alpha) \tag{3.22}$$

This formulation results in the following properties for pure slip conditions

$$\lim_{\kappa\to\infty} F_x = \mu_x F_z C_{ex}, \qquad \lim_{\alpha\to\infty} F_y = \mu_y F_z C_{ey} \tag{3.23}$$

$$C_{fx} = \mu_x F_z(E_{ex} + C_{ex} b_{ex}), \qquad C_{fy} = \mu_y F_z(E_{ey} + C_{ey} b_{ey}) \tag{3.24}$$

$$\kappa_{peak} = \frac{E_{ex} + C_{ex} b_{ex}}{E_{ex} b_{ex}}, \qquad \alpha_{peak} = \frac{E_{ey} + C_{ey} b_{ey}}{E_{ey} b_{ey}} \tag{3.25}$$

Here $C_{fx,y}$ represents the cornering stiffness and longitudinal stiffness at zero slip, and x_{peak} the location of the peak force. (3.23) gives the limit value of tyre force in pure slip conditions.

3.2.3 Magic Formula

The magic formula is an empirical tyre model that describes the tyre characteristics based on tyre measurements in combination with tyre identification algorithms. This model was developed under Prof. H. Pacejka and is extensively described in [28]. TNO has commercialised this model by providing simulation packs for various computational packages such as Matlab Simulink under the name Delft-tyre. The magic formula is widely adopted in vehicle dynamics modelling. The general magic formula for a fixed vertical load and camber angle, with horizontal and vertical shifts omitted reads,

$$F_{x0} = D_x \sin[C_x \arctan\{B_x\kappa - E_x(B_x\kappa - \arctan B_x\kappa)\}] \tag{3.26}$$

$$F_{y0} = D_y \sin[C_y \arctan\{B_y\alpha - E_x(B_y\alpha - \arctan B_y\alpha)\}] \tag{3.27}$$

$$F_x = G_{x\alpha}(\alpha, \kappa) F_{x0}(\kappa, \gamma, F_z) \tag{3.28}$$

$$F_y = G_{y\kappa}(\alpha, \kappa) F_{y0}(\alpha, \gamma, F_z) \tag{3.29}$$

With F_{x0} and F_{y0} being of the form (3.26),(3.27), representing the pure slip characteristics. $G_{x\alpha}(\alpha, \kappa)$ and $G_{y\kappa}(\alpha, \kappa)$ are weighting functions for handling combined slip conditions. Depending on the available measurements, coefficients of (3.26)-(3.29), can be further specified to include vertical load, camber angle and inflation pressure effects.

Similarity Method

In case only very limited measurements are available, the combined slip equations can be reduced using the similarity method as described in [28]. This method uses the insights gained from the brush model to describe combined slip forces using theoretical slips σ_x and σ_y. These theoretical slips are combined to total slip σ via (3.30) which is substituted in the pure slip equations (3.32) (3.31). The combined slip forces are then found by scaling the force with the slip in either x or y direction divided by the total slip via (3.33). Note also the addition of a small value ϵ in (3.30), which prevents numerical issues in (3.33) when total slip approaches zero.

$$\sigma_x = \frac{\kappa}{1+|\kappa|} \qquad \sigma_y = \frac{\tan(\alpha)}{1+|\kappa|} \qquad \sigma = \sqrt{\sigma_x^2 + \sigma_y^2 + \epsilon} \tag{3.30}$$

$$F_{y0}(\sigma) = D_{mf_y} \sin[C_{mf_y} \arctan\{B_{mf_y}\sigma - E_{mf_y}(B_{mf_y}\sigma - \arctan B_{mf_y}\sigma)\}] \tag{3.31}$$

$$F_{x0}(\sigma) = D_{mf_x} \sin[C_{mf_x} \arctan\{B_{mf_x}\sigma - E_{mf_x}(B_{mf_x}\sigma - \arctan B_{mf_x}\sigma)\}] \tag{3.32}$$

$$F_y = F_{y0}\frac{\sigma_y}{\sigma} \qquad F_x = F_{x0}\frac{\sigma_x}{\sigma} \tag{3.33}$$

Although the Magic formula offers a very accurate and direct description of tyre characteristics, some disadvantages exist. This is especially the case when using the magic formula in combination with an EKF based observer. This EKF VSE requires derivation of the expressions with respect to the states. Due to the fairly complex formulation of the full magic formula, calculating these derivatives results in very lengthy expressions. An advantage of the magic formula is that, since the model is widely adopted, tyre parameters are often available for this formulation. This makes implementation for vehicle state estimation faster.

3.2.4 Overview

A comparison of the characteristics of the tyre models is presented in Figure 3.7. Parameters have been chosen such that cornering stiffness and friction are identical for all models. For the magic formula and extended exponential model, the limit value has is also chosen to be equal. It can clearly be seen that despite this selection of parameters, the models are behaving differently near their peak values.

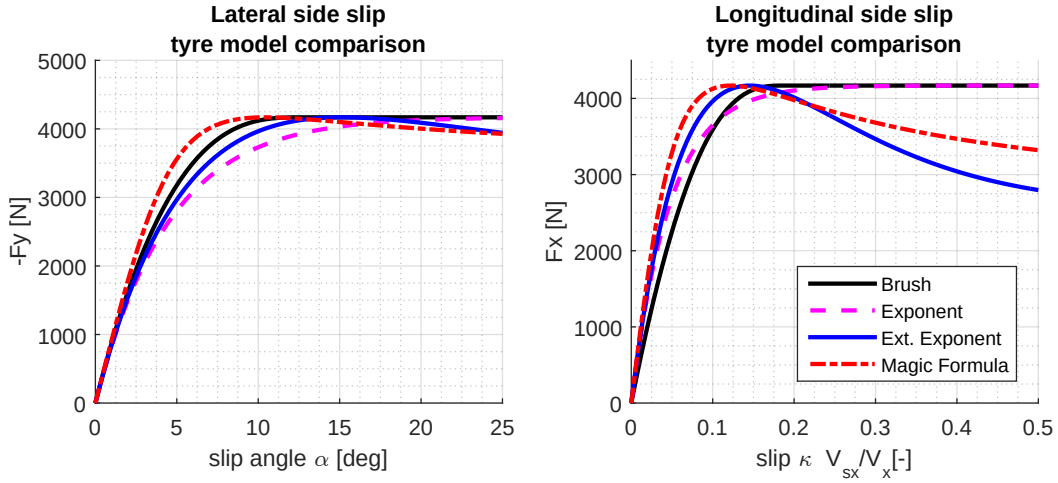


Figure 3.7: Comparison of tyre models, friction and cornering stiffness have been matched across all models.

The magic formula is selected to be applied in the VSE. This is due to more degrees of freedom being available. When desired, parameters in the magic formula can be chosen such that behaviour of the other models is mimicked. Initially, the full set of equations presented in

[28] is integrated in the developed VSE. However, the absence of dedicated tyre tests made this implementation overly ambitious. Therefore, camber effects and pressure effects are neglected and the similarity method is used to further reduce the amount of parameters needed. The final set of equations used as tyre model are presented in appendix B.

A final important remark is the observation that, although the parameter μ is used across various models, its behaviour is rather different. This is important to note, as friction will be included in the VSE as augmented state. This will become important when comparing the obtained results, with those of the VSE already adopted by TNO, which uses the exponential tyre model. The effect of scaling the friction parameter can be seen in Figure 3.8. It is quite clear that for the exponential tyre, not only maximum force, but also cornering stiffness is changed. Note that if desired, the magic formula does allow to mimic this behaviour, by also scaling the cornering stiffness with the same value.

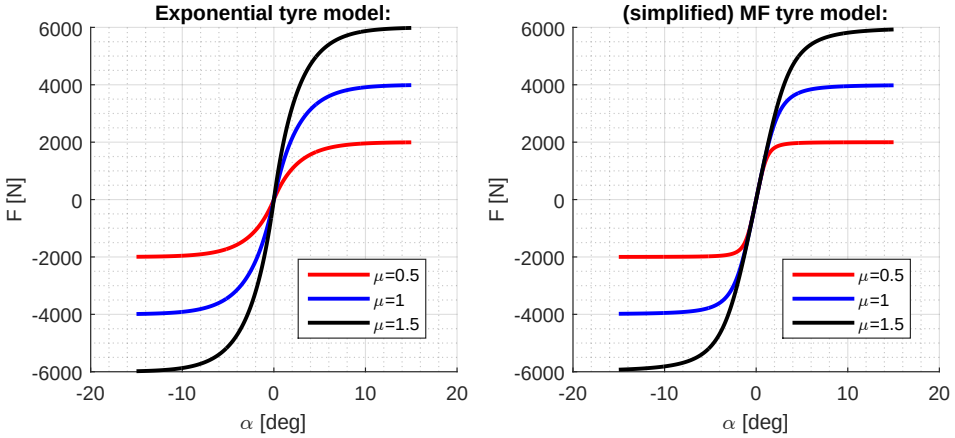


Figure 3.8: Difference in influence of scaling of friction parameter. Note that if desired, the magic formula does allow to mimic this behaviour

3.2.5 Relaxation Length

Due to the noisy signals of the wheelspeeds, the resulting output, a_x, is excessively noisy at low speeds when directly using the equations described in section 3.2.3. This problem is also illustrated in Appendix B.4. To overcome this problem, the relaxation length of the tyre has been modelled using a first order system [31]. More sophisticated descriptions are available and are given in [28]. However, since this simpler approach solves the problem at hand, it is chosen not to add additional complexity. The first order system has a time constant τ which is the relaxation length in that direction divided by the forward velocity.

$$\tau_y \frac{dv_{sy}^*}{dt} + v_{sy}^* = v_{sy} \qquad \tau_y = \frac{\sigma_{rel_x}}{v_x} \tag{3.34}$$

$$\tau_x \frac{dv_{sx}^*}{dt} + v_{sx}^* = v_{sx} \qquad \tau_x = \frac{\sigma_{rel_y}}{v_x} \tag{3.35}$$

The resulting v_{sy}^* and v_{sx}^* are used in the equations for the tyre model as described before.

3.2.6 Friction Estimation

It is desired for control applications to have an estimate of the road friction, as this is crucial in determining the limit handling conditions of the vehicle. For example, the friction estimate can be used to determine if a collision avoidance manoeuvre is feasible. Additionally, the VSE is able to adapt to varying conditions, resulting in more accurate state estimates. To obtain such

an estimate of friction, road friction scaling coefficient λ_μ is appended as an augmented state to the states already described in section 3.1. The friction coefficient calculated in the tyre model μ_0 is thus scaled by the augmented state λ_μ to obtain the actual friction limit for tyre i, μ_i

$$\mu_i = \lambda_\mu \mu_0(F_{z_i}) \tag{3.36}$$

3.3 Parameter Identification and Validation

So far, the both vehicle and tyre model have been presented and discussed. This section will cover the parameter identification process for the model of the Toyota Prius used by TNO. After the parameters have been identified, the model will be validated by using measured data sets, which are generated at the ATP (Automotive Testing Papenburg GmbH) test track.

3.3.1 Parameter Identification

The most basic parameters of the Toyota Prius 2012 can be found rather straightforward. Geometric values of wheelbase and track width are provided by the manufacturer [32]. Corner weights of the car are measured preceding the measurements at ATP. These are then used to determine the position of the center of gravity and vehicle mass. Center of gravity height and inertial properties are estimated using approximate fits presented in [33] and checked by trends in the database presented in [34]. Aerodynamic properties of the vehicle are stated in the product information provided by Toyota [32], with center of pressure height estimated at half the vehicle's height. Roll stiffness distribution is estimated based on similar vehicles measured by TNO. The roll gradient is calculated using the roll rate sensor for a J-turn manoeuvre and compared with values from [27]. The Ackerman ratio is deduced from the Prius workshop manual [35]. The Mechanical trail is estimated from the caster angle in [35]. Front steering compliance is unknown and fitted to the measured data.

Table 3.1: Vehicle parameters: Toyota Prius 2012

parameters	**value**	**unit**	**designation**	**source**
L_{wb}	2.7	$[m]$	wheelbase	[32]
L_{tw_f}	1.519	$[m]$	front track width	[32]
L_{tw_r}	1.509	$[m]$	rear track width	[32]
m_f	966	$[kg]$	mass on front axle	measured
m_r	688	$[kg]$	mass on rear axle	measured
a	1.123	$[m]$	distance of COG to front axle	calculated
b	1.577	$[m]$	distance of COG to rear axle	calculated
I_{zz}	2930	$[kgm^2]$	yaw moment of inertia	estimated [34][33]
h_{cog}	0.54	$[m]$	height of COG (center of gravity)	estimated [34][33]
A_{drag}	2.17	$[m^2]$	area of drag	[32]
C_d	0.25	$[-]$	drag coefficient	[32]
h_{cop}	0.75	$[m]$	height of COP (center of pressure)	estimated
K_ϕ	0.45	$[\deg s^2/m]$	roll gradient	calculated [27]
$\frac{k_{\phi_1}}{k_{\phi_{tot}}}$	0.53	$[-]$	fraction of roll stiffness on front axle	estimated
r_{ack}	0.75	$[-]$	fraction of Ackerman steering	calculated [35]
i_{steer}	15.648	$[-]$	steering ratio	measured
K_{Mz_f}	8e-3	$[\deg/Nm]$	front steering compliance	fitted
K_{Mz_r}	0	$[\deg/Nm]$	rear steering compliance	fitted
L_{trail_f}	0.031	$[m]$	front mechanical trail	estimated [35]

3.3.2 Tyre Parameters

This section will describe the process of finding the tyre parameters used in the model. In absence of dedicated tyre measurements, the magic formula will be used with the similarity method to handle combined slip, as suggested in section 3.2.3. The remaining parameters are given in Table B.1. A method similar to that of [36] is used in order to fit the pure slip parameters. In absence of measured data, aligning moment coefficients have been taken directly from [28]. The compliance parameter is then tuned with these alignment parameters fixed. Although at this point the aligning moment is an unknown, it has been modelled, such that once data becomes available, the model accuracy increases. It should be noted that when such data becomes available, compliance parameter K_{Mz_f}, should be re-tuned.

Longitudinal

Fitting of longitudinal parameters is performed on a manoeuvre with pure longitudinal excitation and maximum (ABS) braking. Identification of parameters P_{DX1}, P_{KX1} and P_{EX1} is performed by first finding the maximum acceleration in Figure 3.9. This identifies P_{DX1}, as P_{DX1} describes the maximum normalized tyre force. P_{KX1} is then found by fitting the acceleration zone between $t = 2s-10s$. Finally P_{EX1} is tuned to minimize the overall error in both acceleration and velocity.

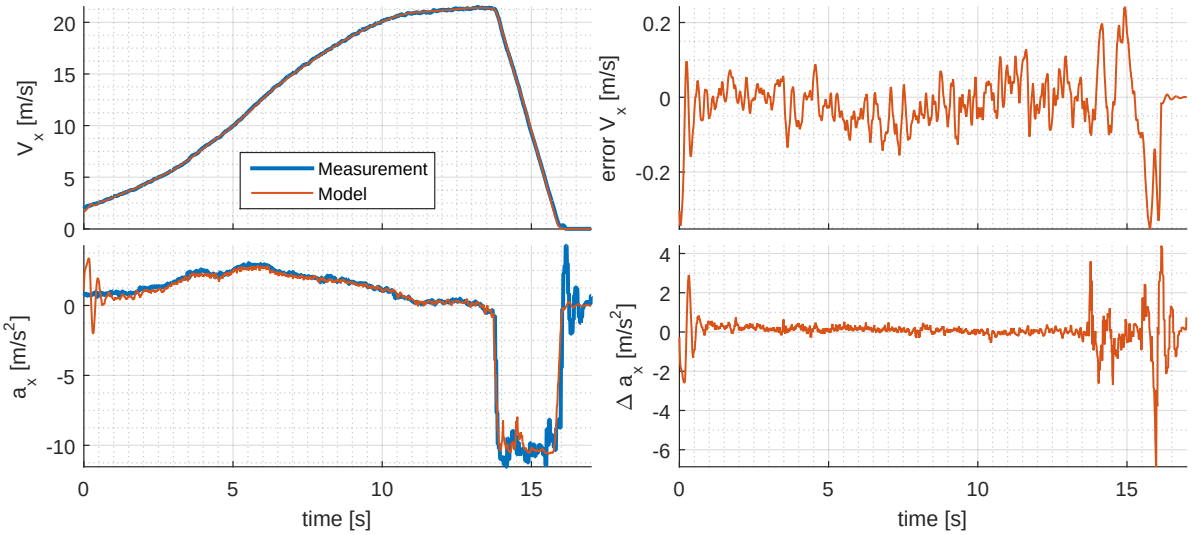


Figure 3.9: Fitting results of longitudinal parameters P_{DX1}, P_{KX1} and P_{EX1} on accelerating and maximum braking manoeuvre

Lateral

Lateral coefficients are tuned to quasi-steady-state circular tests. Smaller lateral accelerations are tuned by cornering stiffness factor P_{KY1} and front axle compliance K_{Mz_f}. Higher lateral loads are tuned by parameters for maximum friction P_{DY1} and curvature parameter P_{EY1}. Results can be found in Figure 3.10. Note the higher measured lateral acceleration with respect to the model. This can be explained by the fact that the sensor for a_y is mounted on a rolling body. Due to roll of this body, an additional component, $g \sin \phi$, containing the gravitational acceleration, is measured. This is compensated in the model by calculating the steady-state roll angle, and including the resulting gravity component at that steady-state angle. The roll gradient is estimated from values of [27]. Note that in Figure 3.10, at $t \approx 25s$ heavy braking is applied while cornering. These combined slip conditions are accurately captured by the similarity assumption described in section 3.2.

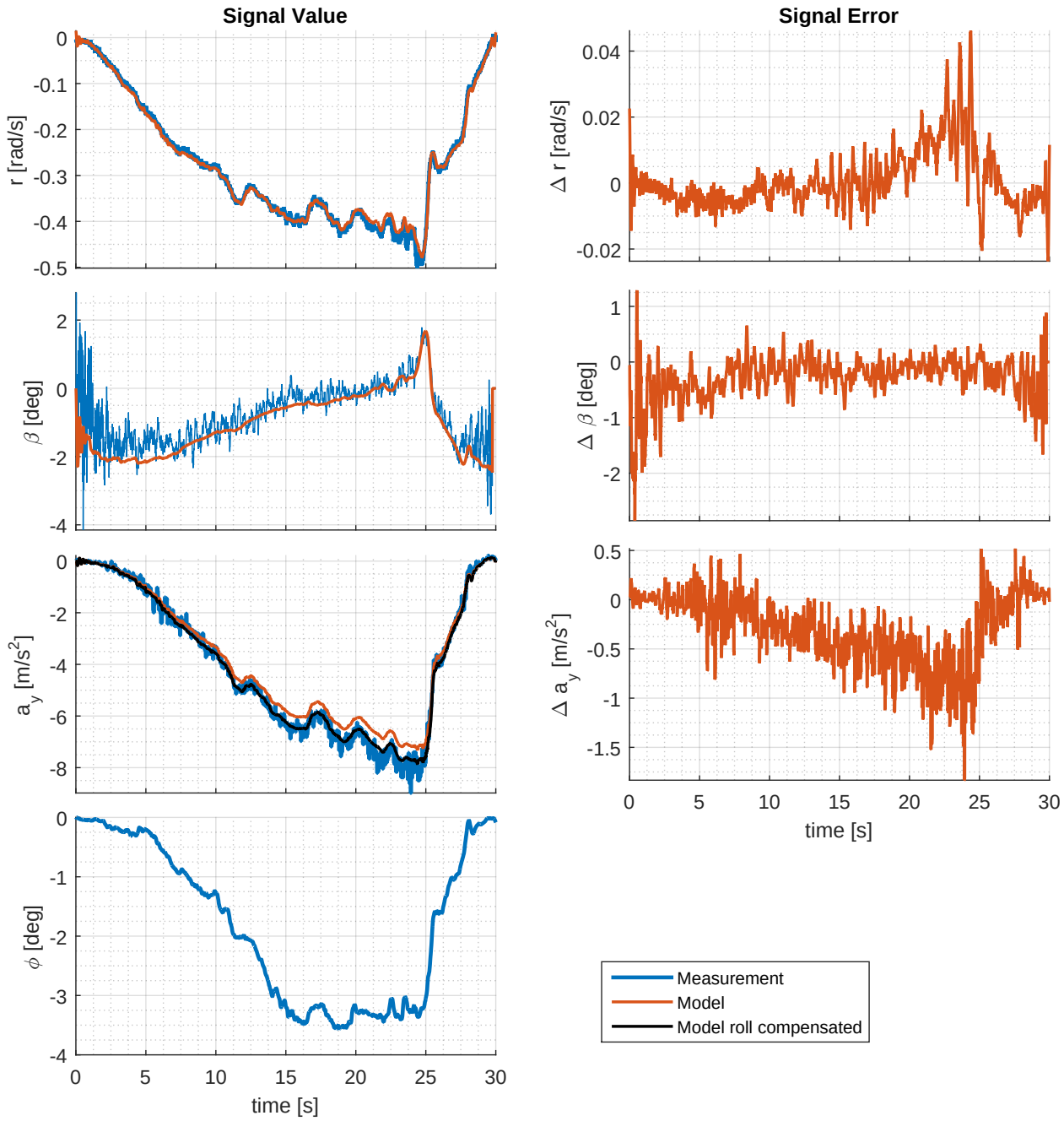


Figure 3.10: Results of fitting the lateral parameters P_{DY1}, P_{KY1}, P_{EY1} and K_{Mz_f} on quasi-steady-state circular test. Note the steady-state roll angle compensation for the model in a_y.

3.3.3 Validation

With the parameters fitted, additional tests are performed to validate the model. Especially of interest are manoeuvres with more dominant transient dynamics. Standardized tests like lane changes and J-turns are used. Figure 3.11 displays results of such a lane change manoeuvre. It can be seen that state errors across all states are relatively small.

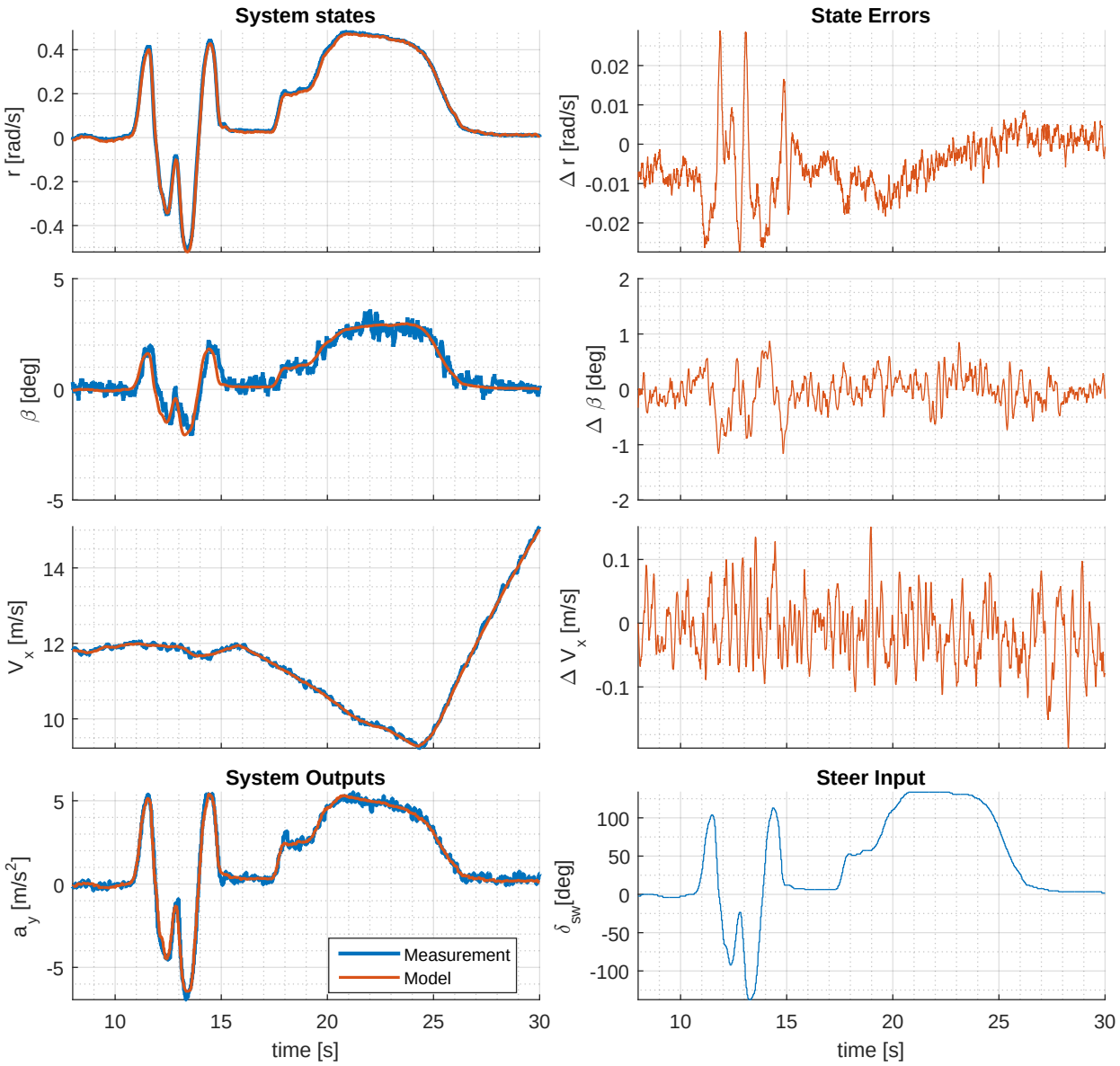


Figure 3.11: Model results for lane change manoeuvre

3.4 Summary

In this chapter, the vehicle model that is used for the vehicle state estimator is described. The model choice is motivated using a brief analysis that has shown that all relevant dynamics are described well by the two track model. All equations of motion and other relevant equations are presented such that easy reproduction of results is possible. Additionally, the several options for the tyre model are briefly discussed. The benefits of the magic formula are briefly discussed, which motivates the use of this model in the vehicle state estimator. Lack of tyre measurements is partly solved by simplification of the combined slip characteristics using the similarity method. Vehicle parameters and how they are obtained are presented. Parameter identification of tyre parameters is presented and supported with fitted results. The resulting vehicle model is demonstrated to accurately describe both longitudinal as lateral vehicle behaviour.

Chapter 4

State Estimator Implementation

Chapters 2 and 3 have discussed the theory regarding observers and vehicle modelling respectively. Implementation of the vehicle model for application in the Vehicle State Estimator is discussed here. The derivations of matrix valued functions $\mathbf{A}(\mathbf{x}, \mathbf{u})$ and $\mathbf{H}(\mathbf{x}, \mathbf{u})$, that describe the dynamics and measurements, are presented. Also the additional logic that is implemented in parallel to the estimator to aid in the vehicle state estimator will be discussed. Finally, some considerations for evaluating the performance of the implemented estimator will be presented.

4.1 Model Outputs

The estimator algorithms requires measurable outputs on which the state corrections are based. Typically, an Inertial Measurement Unit (IMU) is available in the vehicle for Electronic Stability Control purposes. These IMUs consist of accelerometers and gyroscopes, which measure the linear accelerations and rotational velocities. Additionally, the vehicle's sensors provides additional measurement of the vehicle, which can include wheelspeed signals and GPS signals. The yaw rate of the vehicle is measurable directly via the IMU. The remaining outputs are given by the following relations.

$$a_x = \sum_{i=1}^{4} \frac{F_{cx_i}}{m} \tag{4.1}$$

$$a_y = \sum_{i=1}^{4} \frac{F_{cy_i}}{m} \tag{4.2}$$

$$V_{\text{gps}} = \sqrt{V_x^2 + V_y^2} \tag{4.3}$$

$$V_{x_{\text{rear}}} = \frac{1}{2} \sum_{i=3}^{4} (\omega_i R_{e_i} + v_{sx_i}) \tag{4.4}$$

It should be noted that once roll and pitch dynamics are included, or road bank angle and road inclination are considered. The measured accelerations will be different then represented here, due to a component of gravitation acceleration being measured. This was already discussed briefly in Section 3.3.2, where it was demonstrated how the measured lateral acceleration was corrected using a steady state roll angle value. More details can be found in Appendix D.1.2. Both road inclination and road bank angle could be accounted for by estimating them as external disturbances. It should be noted that this simultaneous estimation is significantly more difficult, since friction, road bank angle and body slip angle are all corrected primarily via the lateral acceleration.

As can be seen, multiple measurements are taken related to velocity. In cases where longitudinal slip is small, (4.4) can be used to find the forward velocity. When a front wheel driven vehicle is assumed, significant slip is primarily encountered during braking. The only force generated for driving, is that of rolling resistance and accelerating the rear wheels. These forces are however small, and hence slips are also small. For small slips, (4.4) results directly in the vehicle state V_x, since the rear wheels are not steered. Unlike (4.4), (4.3) is still capable of providing accurate measurements under hard braking. However, GPS signals can suffer delays and accuracy depends on availability of satellites. Combining (4.4) and (4.3) can therefore be used to provide reliable velocity information.

4.2 Filter Gains

Both the EKF and SDRE algorithm require the calculation of filter gain $\mathbf{K}(\mathbf{x})$. This section will describe the process of calculating these values online. As described in Chapter 2, calculation of the filter gain $\mathbf{K}(\mathbf{x})$ requires the matrix valued descriptions of dynamics and measurements $\mathbf{A}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ respectively, as given by (2.14) and (2.15). These can be obtained either through linearisation, as is the case for the EKF, or through factorisation, as is the case for the SDRE estimator.

4.2.1 Extended Kalman Filter

The Jacobian matrices of state updates and system outputs, (2.18) and (2.19), are required for the EKF. To facilitate easy implementation of different tyre models, it is convenient to isolate the effects of the non-linear tyre in the Jacobian matrices. Consider the description of the planar two track vehicle model given in Section 3.1 with the addition augmented friction state λ_μ, such that $\mathbf{x} = \begin{bmatrix} r & V_y & V_x & \lambda_\mu \end{bmatrix}^T$. Inputs are the steering angle and wheelspeed signals $\mathbf{u} = \begin{bmatrix} \delta & \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$ that directly effect the forces generated at each wheel.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{r} \\ \dot{V}_y \\ \dot{V}_x \\ \dot{\lambda}_\mu \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} \frac{M_{z_i} + d_{y_i} F_{cx_i} + d_{x_i} F_{cy_i}}{I_z} \\ \sum_{i=1}^{4} \frac{F_{cy_i}}{m} \\ \sum_{i=1}^{4} \frac{F_{cx_i}}{m} - \frac{F_{\text{drag}}}{m} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -rV_x \\ rV_y \\ 0 \end{bmatrix} \tag{4.5}$$

This is identical to the model given in Section 3.1, by (3.1), but with λ_μ the augmented state that represents the friction scaling parameter. The tyre forces and moments in chassis coordinates are found by pre multiplying the tyre forces in wheel plane coordinates with a rotation matrix using the wheel angle δ_i via

$$\mathbf{F}_{c_i} = \begin{bmatrix} F_{cx_i} \\ F_{cy_i} \\ M_{z_i} \end{bmatrix} = \begin{bmatrix} \cos\delta_i & -\sin\delta_i & 0 \\ \sin\delta_i & \cos\delta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{F}_{\text{tyre}_i} \tag{4.6}$$

$$\mathbf{F}_{\text{tyre}_i} = \begin{bmatrix} F_{x_i} & F_{y_i} & M_{z_i} \end{bmatrix}^T \tag{4.7}$$

The tyre can be considered a subsystem of the total vehicle with contact patch states and output forces as described for tyre i by (4.7). Since the state equations contain only linear contributions

of these tyre forces, the vehicle state equations (4.5) can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} \dot{r} \\ \dot{V}_y \\ \dot{V}_x \\ \dot{\lambda}_\mu \end{bmatrix} = \mathbf{J}_1 \mathbf{F}_{\text{tyre}} + \begin{bmatrix} 0 \\ -rV_x \\ rV_y \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \frac{C_{d_{\text{eff}}} V_x^2}{2m} \\ 0 \end{bmatrix}, \qquad \mathbf{J}_1(\delta) = \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{F}_{\text{tyre}}} \tag{4.8}$$

$$\mathbf{F}_{\text{tyre}} = \begin{bmatrix} \mathbf{F}_{\text{tyre}_1}^T & \mathbf{F}_{\text{tyre}_2}^T & \mathbf{F}_{\text{tyre}_3}^T & \mathbf{F}_{\text{tyre}_4}^T \end{bmatrix}^T \tag{4.9}$$

Where $\mathbf{F}_{\text{tyre}}$ is a column containing all forces and moments generated by all four tyres, as is given for a single tyre by (4.7). Note that the matrix $\mathbf{J}_1$ only depends on inertial and geometric parameters and steering angles δ_i. Note also that $\mathbf{F}_{\text{tyre}}$ is again a function of the tyre states $\mathbf{x}_{\text{tyre}}$, as described in Section 3.2.3. This relation is determined by the type of model that is used. Hence, isolating these parts of the Jacobian is desired, to easily replace and compare tyre models. Derivation of the Jacobian $\mathbf{A}$ leads to

$$\begin{aligned} \mathbf{A} = \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}} &= \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} \dot{r} \\ \dot{V}_y \\ \dot{V}_x \\ \dot{\lambda}_\mu \end{bmatrix} = \frac{\partial}{\partial \mathbf{x}} \left(\mathbf{J}_1 \mathbf{F}_{\text{tyre}} \right) + \frac{\partial}{\partial \mathbf{x}} \left(\begin{bmatrix} 0 \\ -rV_x \\ rV_y \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \frac{C_{d_{\text{eff}}} V_x^2}{2m} \\ 0 \end{bmatrix} \right) \\ &= \left(\mathbf{J}_1 \mathbf{J}_{\text{tyre}} \frac{\partial \mathbf{x}_{\text{tyre}}}{\partial \mathbf{x}} \right) + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -V_x & 0 & -r & 0 \\ V_y & r & -\frac{C_{d_{\text{eff}}} V_x}{m} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{J}_{\text{tyre}} = \frac{\partial \mathbf{F}_{\text{tyre}}}{\partial \mathbf{x}_{\text{tyre}}} \end{aligned} \tag{4.10}$$

where $\mathbf{J}_1$ is the same matrix described in (4.8) and where $\mathbf{x}_{\text{tyre}}$ is a column that contains the contact patch states of all tyres

$$\mathbf{x}_{\text{tyre}} = \begin{bmatrix} \mathbf{x}_{\text{tyre}_1}^T & \mathbf{x}_{\text{tyre}_2}^T & \mathbf{x}_{\text{tyre}_3}^T & \mathbf{x}_{\text{tyre}_4}^T \end{bmatrix}^T, \qquad \mathbf{x}_{\text{tyre}_i} = \begin{bmatrix} v_{sx_i} & v_{sy_i} & v_{cx_i} & \lambda_\mu \end{bmatrix}^T \tag{4.11}$$

These tyre states, $\mathbf{x}_{\text{tyre}_i}$, are a function of vehicle states $\begin{bmatrix} r & V_y & V_x & \lambda_\mu \end{bmatrix}^T$ and system inputs $\begin{bmatrix} \delta_i & \omega_i \end{bmatrix}^T$. They can be found by calculating the speeds at wheel positions in chassis coordinates and rotating about wheel angle δ_i

$$\mathbf{x}_{\text{tyre}_i} = \mathbf{J}_{2_i} \begin{bmatrix} r \\ V_y \\ V_x \\ \lambda_\mu \end{bmatrix} + R_w \begin{bmatrix} \omega_i \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{J}_{2_i} = \begin{bmatrix} d_{y_i} \cos\delta_i - d_{x_i} \sin\delta_i & -\sin\delta_i & \cos\delta_i & 0 \\ d_{y_i} \sin\delta_i + d_{x_i} \cos\delta_i & \cos\delta_i & \sin\delta_i & 0 \\ d_{y_i} \cos\delta_i - d_{x_i} \sin\delta_i & -\sin\delta_i & \cos\delta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

Herein, d_{x,y_i} represent the components of the position vector of each wheel with respect to the vehicles center of gravity in the chassis fixed frame. Derivation of the term $\frac{\partial \mathbf{x}_{\text{tyre}}}{\partial \mathbf{x}}$ follows directly. It should be noted that the only term that is affected by the choice of tyre model in (4.10), is the $\mathbf{J}_{\text{tyre}}$ term. By isolating this term, the used tyre model can conveniently be changed if so desired.

$$\mathbf{A} = \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}} = (\mathbf{J}_1 \mathbf{J}_{\text{tyre}} \mathbf{J}_2) + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -V_x & 0 & -r & 0 \\ V_y & r & -\frac{C_{d_{\text{eff}}} V_x}{m} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{J}_2 = \begin{bmatrix} \mathbf{J}_{2_1} \\ \mathbf{J}_{2_2} \\ \mathbf{J}_{2_3} \\ \mathbf{J}_{2_4} \end{bmatrix} \tag{4.13}$$

Herein, $\mathbf{J}_2$ is the matrix resulting from combining the matrices $\mathbf{J}_{2_i}$ (given in (4.12)), and represents the transformation of vehicle states to tyre states. Derivation of the output Jacobian $\mathbf{H}(\mathbf{x})$ follows the same steps and will not be further discussed.

Combined slip Tyre Jacobian Matrix

The expression for the tyre Jacobian $\mathbf{J}_{\text{tyre}}$ is initially derived symbolically using Matlab, for the full set of equations presented in [28]. Symbolic substitution is used to simplify these expressions in order to speed up simulation times. Calculated tyre Jacobians are validated using dedicated tests. Later, it was concluded that the total set of equations is far too detailed for the intended purpose of state estimation. Moreover, obtaining the additional parameter requires dedicated measurements, which are not available. This leads to the change of the tyre model described in Section 3.2.3. Although the remainder of this thesis will only use this reduced model, it is worth to note that computational effort using the full set of equations and the resulting tyre Jacobian is significantly larger (up to factor 3). This statement holds for both compilation time and the execution time of the Simulink simulation.

4.2.2 State Dependent Riccati Equation

As mentioned in Section 2.1, the SDRE estimator involves the construction of a State-Dependent Coefficient (SDC), instead of the Jacobian in the EKF. This section will cover the criteria involved an SDC form to be valid, and the derivation of a set of valid SDC forms that are used in the calculation of the filter gains.

State Dependent Coefficient construction

Calculation of the SDC form follows a very similar method as described for the EKF in the sense that tyre effects have been isolated. However, the tyre Jacobian $\mathbf{J}_{\text{tyre}}$ presented in (4.10) is replaced by an SDC form $\mathbf{A}_{\text{tyre}_i} \in \mathbb{R}^{3\times 4}$. The tyre subsystem $\mathbf{F}_{\text{tyre}_i} = \mathbf{f}_{\text{MF}_i}(\mathbf{x}_{\text{tyre}_i})$ is therefore factorised as

$$\mathbf{F}_{\text{tyre}_i} = \mathbf{A}_{\text{tyre}_i}(\mathbf{x}_{\text{tyre}_i})\mathbf{x}_{\text{tyre}_i} \tag{4.14}$$

Many valid solutions exist for this SDC form, as described in Section 2.1.5. However, all elements in $\mathbf{A}_{\text{tyre}_i}$, denoted as a_{ij}, should be finite. This chapter will only discuss the SDC form implemented for the final estimator.

For the linear case, the Kalman filter is the optimal estimator. Considering this, it makes sense to make sure that, for tyres operating in the linear regime, the coefficients used the SDRE reduces to the linear tyre description. To this extent, the lateral tyre force should reduce to a product of cornering stiffness and slip angle. Similarly, longitudinal force should reduce to the product of slip stiffness and slip ratio for low slips. Consider the following factorisation of lateral and longitudinal tyre force

$$\begin{aligned} F_{x_i} &= \frac{f_x(\mathbf{x}_{\text{tyre}_i})}{v_{sx_i}}\left(1-\frac{v_{sy_i}}{v_{x_i}}\right)v_{sx_i} + \frac{f_x(\mathbf{x}_{\text{tyre}_i})}{v_{x_i}}v_{sy_i} \\ F_{y_i} &= \frac{f_y(\mathbf{x}_{\text{tyre}_i})}{v_{x_i}}v_{sx_i} + \frac{f_y(\mathbf{x}_{\text{tyre}_i})}{v_{sy_i}}\left(1-\frac{v_{sx_i}}{v_{x_i}}\right)v_{sy_i} \\ M_{z_i} &= \frac{f_m(\mathbf{x}_{\text{tyre}_i})}{v_{x_i}}v_{sx_i} + \frac{f_m(\mathbf{x}_{\text{tyre}_i})}{v_{sy_i}}\left(1-\frac{v_{sx_i}}{v_{x_i}}\right)v_{sy_i} \end{aligned} \quad \Rightarrow \overbrace{\begin{bmatrix} F_{x_i} \\ F_{y_i} \\ M_{z_i} \end{bmatrix}}^{\mathbf{F}_{\text{tyre}_i}} = \overbrace{\begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 \end{bmatrix}}^{\mathbf{A}_{\text{tyre}_i}} \overbrace{\begin{bmatrix} v_{sx_i} \\ v_{sy_i} \\ v_{cx_i} \\ \lambda_\mu \end{bmatrix}}^{\mathbf{x}_{\text{tyre}_i}} \tag{4.15}$$

Where f_x, f_y and f_m represent the tyre model functions for longitudinal and lateral forces and self aligning moment. For small slip it can be shown for (4.15) that for the individual elements it holds that

$$
\begin{aligned}
&\lim_{v_{sx,y}\to 0} \frac{f_{x,y,m}(\mathbf{x}_{\text{tyre}_i})}{v_{x_i}} && = 0 \\
&\lim_{v_{sx,y}\to 0} \frac{f_x(\mathbf{x}_{\text{tyre}_i})}{v_{sx_i}}\left(1-\frac{v_{sy_i}}{v_{x_i}}\right) = \frac{C_{\kappa x}}{v_{x_i}} \quad \Rightarrow \quad \lim_{v_{sx}\to 0} C_{\kappa x}\kappa && = C_{\kappa x}\frac{v_{sx_i}}{v_{x_i}} \\
&\lim_{v_{sx,y}\to 0} \frac{f_y(\mathbf{x}_{\text{tyre}_i})}{v_{sy_i}}\left(1-\frac{v_{sx_i}}{v_{x_i}}\right) = \frac{C_{\alpha y}}{v_{x_i}} \quad \Rightarrow \quad \lim_{v_{sy}\to 0} C_{\alpha y}\alpha && = C_{\alpha y}\frac{v_{sy_i}}{v_{x_i}} \\
&\lim_{v_{sx,y}\to 0} \frac{f_m(\mathbf{x}_{\text{tyre}_i})}{v_{sx_i}}\left(1-\frac{v_{sy_i}}{v_{x_i}}\right) = \frac{C_{\alpha y}L_{\text{pt}_0}}{v_{x_i}} \quad \Rightarrow \quad \lim_{v_{sy}\to 0} C_{\alpha y}\alpha L_{\text{pt}_0} && = C_{\alpha y}L_{\text{pt}_0}\frac{v_{sy_i}}{v_{x_i}}
\end{aligned}
\tag{4.16}
$$

Where L_{pt_0} represents the nominal pneumatic trail. Equation (4.16) shows that for small slip the factorisation reduces to the linear tyre representation, such that the desired relation for low slip is achieved. Additionally, it can be seen that in combined slip conditions the element that represents the cornering stiffness, a_{22}, reduces for increasing longitudinal slips. Similarly, the coefficient representing the slip stiffness, a_{11}, decreases for increasing lateral slip velocities. This effect can also be seen on the actual tyre, as also illustrated in Appendix B.

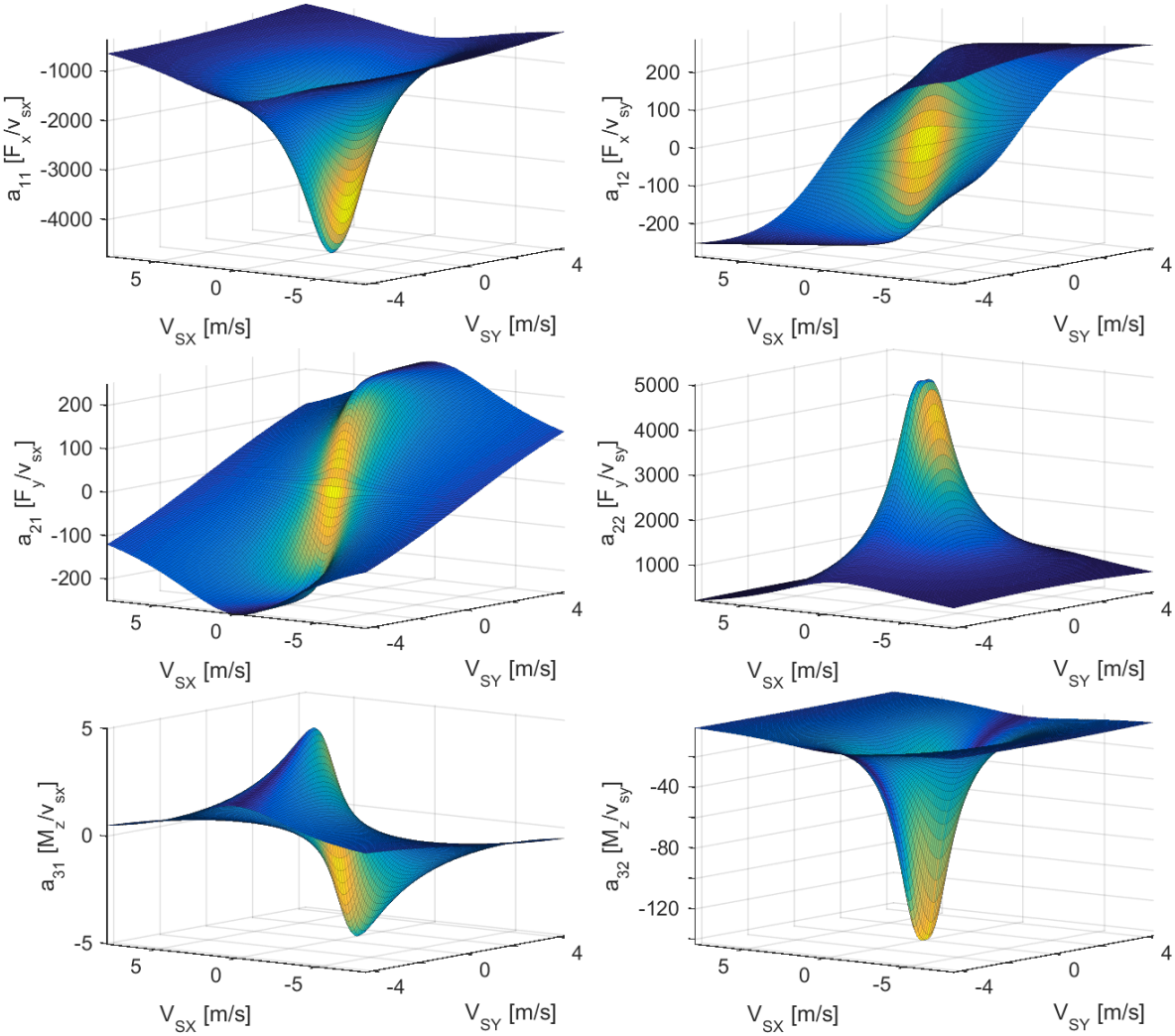


Figure 4.1: Non-zero elements of SDC form (4.15) for $v_x = 16[m/s]$ and $F_z = 4000[N]$. Forces are factorised as: $F_x = a_{11}v_{sx} + a_{12}v_{sy}$, $F_y = a_{21}v_{sx} + a_{22}v_{sy}$, $M_z = a_{31}v_{sx} + a_{32}v_{sy}$. Colors indicate magnitude of the gradient.

The limit values described in (4.16) could be used to describe the coefficients for zero slip. However, the expression for these limit values are computational expensive. Instead, a small lower bound is placed on the absolute value of slip for implementation, such that forces are non-zero and the described limit values (4.16) can be approximated numerically. Figure 4.1 shows the resulting non-zero elements of the SDC form (4.15). It can be seen that the resulting surfaces provide a smooth factorisation.

The first form (4.15) does not show any influence of friction on the developed force. Therefore the estimator would never be able to adjust the friction estimate. By taking as reference the Jacobian $\mathbf{J}_{\text{tyre}}$ used in the EKF, it is observed that the elements corresponding to the friction scaling are only high for high excitations. It is reasonable to assume that something similar is desired in the SDRE estimator. Therefore a second SDC form is used where the force is only factorised by the friction parameters

$$\begin{aligned} F_{x_i} &= \frac{f_x(\mathbf{x}_{\text{tyre}_i})}{\lambda_\mu}\lambda_\mu \\ F_{y_i} &= \frac{f_y(\mathbf{x}_{\text{tyre}_i})}{\lambda_\mu}\lambda_\mu \\ M_{z_i} &= \frac{f_m(\mathbf{x}_{\text{tyre}_i})}{\lambda_\mu}\lambda_\mu \end{aligned} \Rightarrow \begin{bmatrix} F_{x_i} \\ F_{y_i} \\ M_{z_i} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & a_{14} \\ 0 & 0 & 0 & a_{24} \\ 0 & 0 & 0 & a_{34} \end{bmatrix} \begin{bmatrix} v_{sx_i} \\ v_{sy_i} \\ v_{cx_i} \\ \lambda_\mu \end{bmatrix} \tag{4.17}$$

The two presented forms can also easily be applied to the factorisation of the self aligning moment M_z. The combined factorisation of F_x, F_y and M_z results in tyre factorisations $\mathbf{A}_{1_{\text{tyre}_i}}$ from (4.15) and $\mathbf{A}_{2_{\text{tyre}_i}}$ from (4.17) for tyre i. These forms for the individual tyre are then combined for all tyres to $\mathbf{A}_{1,2_{\text{tyre}}} \in \mathbb{R}^{12\times 16}$

$$\mathbf{A}_{2_{\text{tyre}}} = \begin{bmatrix} \mathbf{A}_{2_{\text{tyre}_1}} & & & \\ & \mathbf{A}_{2_{\text{tyre}_2}} & & \\ & & \mathbf{A}_{2_{\text{tyre}_3}} & \\ & & & \mathbf{A}_{2_{\text{tyre}_4}} \end{bmatrix}, \quad \mathbf{A}_{1_{\text{tyre}}} = \begin{bmatrix} \mathbf{A}_{2_{\text{tyre}_1}} & & & \\ & \mathbf{A}_{2_{\text{tyre}_2}} & & \\ & & \mathbf{A}_{2_{\text{tyre}_3}} & \\ & & & \mathbf{A}_{2_{\text{tyre}_4}} \end{bmatrix} \tag{4.18}$$

These two forms can then be combined to a blended form using a parametrisation variable $\boldsymbol{\eta}_{\text{tyre}}$. This parametrisation variable can then be interpreted as a weight that is assigned to the friction state.

$$\mathbf{A}_{\text{tyre}} = \boldsymbol{\eta}_{\text{tyre}}\mathbf{A}_{2_{\text{tyre}}} + (\mathbf{I_{12}} - \boldsymbol{\eta}_{\text{tyre}})\mathbf{A}_{1_{\text{tyre}}} \tag{4.19}$$

The cross-terms in (4.8), $-rV_x$ and rV_y, can also be factorised. These elements have to obvious factorisations, since each element can be factorised by either of its components. Similarly, the drag term is also factorised.

$$\mathbf{A}_{\text{cross}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -r & 0 \\ V_y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{A}_{\text{drag}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{C_{d_{\text{eff}}}V_x}{2m} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.20}$$

By trial and error, the option given by (4.20) is found to work better in all tested conditions. The total factorisation $\mathbf{A}$ that is used in the estimator algorithm is then calculated by (4.21), where $\mathbf{J}_2$ is again defined as in (4.13)

$$\mathbf{A} = \mathbf{J}_1 \left(\boldsymbol{\eta}_{\text{tyre}}\mathbf{A}_{1_{\text{tyre}}} + (\mathbf{I} - \boldsymbol{\eta}_{\text{tyre}})\mathbf{A}_{2_{\text{tyre}}}\right)\mathbf{J}_2 + \mathbf{A}_{\text{cross}} + \mathbf{A}_{\text{drag}} \tag{4.21}$$

parametrisation variable $\boldsymbol{\eta}_{\text{tyre}}$

As briefly mentioned above, two SDC forms, $\mathbf{A}_{1_{\text{tyre}}}$ (4.15) and $\mathbf{A}_{2_{\text{tyre}}}$ (4.17), are created with the intention of modifying of correction that can be applied to the friction state at a given instance. Logically, changes in λ_μ can only be observed if a distinction can be made in the outputs generated by the system. In the EKF, this is implemented by the gradient of tyre forces with respect to λ_μ. Intuitively, one could consider to base the selection of parametrisation variable $\boldsymbol{\eta}_{\text{tyre}}$ using a similar method. Again, multiple functions can be chosen to achieve such a relation where $\boldsymbol{\eta}_{\text{tyre}}$ is related to the amount of excitation of the tyre.

The selected implementation is seen in (4.22), in which $\boldsymbol{\eta}_i$ is proportional to the fraction of the generated force divided by the maximum potential tyre force. This means that if the tyre is excited to its maximum potential, the total factorisation $\mathbf{A}_{\text{tyre}}$, is determined solely by the friction state via $\mathbf{A}_{2_{\text{tyre}}}$. In contrast, when excitation is zero, the factorisation is determined solely via $\mathbf{A}_{1_{\text{tyre}}}$.

$$\boldsymbol{\eta}_{\text{tyre}} = \begin{bmatrix} \boldsymbol{\eta}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{\eta}_2 & \ddots & \vdots \\ \vdots & \ddots & \boldsymbol{\eta}_3 & 0 \\ 0 & \dots & 0 & \boldsymbol{\eta}_4 \end{bmatrix}, \qquad \boldsymbol{\eta}_i = \mathbf{I_3}\eta_i, \qquad \eta_i = \frac{\sqrt{F_{y_i}^2 + F_{x_i}^2}}{F_{z_i}\lambda_\mu} \tag{4.22}$$

Output matrix $\mathbf{H}(\mathbf{x}, \mathbf{u})$

For the factorisation of $\mathbf{h}(\mathbf{x}, \mathbf{u})$, the relation between the accelerations and state update equations is used in the way explained below. Consider the following relations for outputs a_x and a_y

$$a_x = \dot{V}_x - V_y r \tag{4.23}$$

$$a_y = \dot{V}_y + V_x r \tag{4.24}$$

Since factorisations for $\dot{V}_x$ and $\dot{V}_x$ are already generated via (4.21), they can be utilised for the factorisation of a_x and a_y. The third and second row of $\mathbf{A}(\mathbf{x}, \mathbf{u})$ are used to generate the first and second row of $\mathbf{H}(\mathbf{x}, \mathbf{u})$ respectively via

$$\mathbf{H} = \begin{bmatrix} 0 & -r & 0 & 0 \\ 0 & 0 & r & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{A}(\mathbf{x}, \mathbf{u}) \tag{4.25}$$

Additionally in cases when GPS velocity is used instead (or in addition to) the undriven rear wheelspeeds the following factorisation can be used. This will be used during some parts of the simulation, but is not adopted for the experimental part of this project. When an all wheel drive vehicle is assumed and the wheelspeeds can not be used as a velocity measurement, the fourth row of $\mathbf{H}(\mathbf{x}, \mathbf{u})$, denoted as $\mathbf{H}_{4j}$, is replaced by

$$\mathbf{H}_{4j} = \begin{bmatrix} 0 & \frac{\sqrt{V_x^2+V_y^2}}{V_x^2+V_y^2}V_y & \frac{\sqrt{V_x^2+V_y^2}}{V_x^2+V_y^2}V_x & 0 \end{bmatrix} \tag{4.26}$$

4.3 Additional Logic

This section will cover any additional logic involved in implementation of the estimator. This logic is applied to both the EKF and SDRE algorithms. The reason for using this logic and how it was implemented will be discussed here.

4.3.1 Riccati Equation Solver

Usage of the filtering algorithms requires numerical integration of the DRE (2.15). When simple integration of all elements in P is implemented, numerical issues result in the matrix $\mathbf{P}$ no longer being symmetric. Additionally, cases can occur where the diagonal elements in $\mathbf{P}$ become negative, violating the positive definiteness of the error covariance matrix. Both these observations pose a problem for the stability of the algorithm. A possible solution is suggested in [25], making use of the property that $\mathbf{P}$ is symmetric. However, here a simpler solution is adopted. In the integration step a lower bound is placed on diagonal elements in $\mathbf{P}$, ensuring that none can become negative. After numerical integration the following is applied.

$$\mathbf{P} = \frac{\mathbf{P} + \mathbf{P^T}}{2} \tag{4.27}$$

This step prevents accumulation of rounding errors, that would normally result in further deterioration of a numerical non-symmetric matrix $\mathbf{P}$.

4.3.2 Low Speeds Feedback Toggle

The accuracy of the model deteriorates as the forward velocity V_x decreases. This is partly solved by including the relaxation length first order dynamics as described in Section 3.2.3. To prevent the estimator from showing undesired behaviour, a lower bound is placed on the velocity for which the estimator is allowed to correct the states based on measured system outputs. Under that conditions, integration of $\mathbf{P}$ is also halted, since no updates are performed. At standstill all states (with the exception of friction) are fully known. As a result, $\mathbf{P}$ is set to its initial value $\mathbf{P}_0$. This also ensures that the filter behaves identically when starting from standstill when no prior knowledge and information about cross-correlation is available.

4.3.3 Innovation Logic

Some additional manipulation to the innovation signal is done to prevent undesired behaviour of the filter.

Velocity correction under braking

As mentioned in Section 4.1, under braking the velocity measurement performed by the rear wheels becomes unreliable due to wheel slip. To circumvent this problem, the innovation term ν_{V_x} is set to zero if the longitudinal acceleration drops below a braking threshold. This prevents misinformation of the velocity measurement from resulting in wrong corrections.

$$\nu_{V_x} = \begin{cases} \frac{1}{2}\sum_{i=3}^{4}(\omega_i R_{e_i}) - \hat{V}_x & \text{if } a_{x_{\text{meas}}} \geq \frac{F_{\text{drag}}}{m} \\ 0 & \text{if } a_{x_{\text{meas}}} < \frac{F_{\text{drag}}}{m} \end{cases} \tag{4.28}$$

Acceleration Threshold

Although seemingly trivial, steady-state driving still is an important condition. Implementation of the vehicle state observer for road cars will regularly encounter steady-state (highway) driving. A simulation is performed to test this functionality. White noise is added to the wheel speeds signals used in the estimator to more realistically represent the true vehicle. It can be clearly seen in Figure 4.2 that both the EKF and SDRE estimator show drift in the λ_μ estimate. This can be explained by considering the noise band of the longitudinal acceleration. The larger magnitude in acceleration implies a larger tyre force, such that the friction estimated is steered

towards a lower value to correct for this seeming error. The friction estimate uses the difference in magnitude of the accelerations, rather than simply the difference. From Figure 4.2, it can be seen that due to the wheelspeed noise, the magnitude of the noise band in a_x is persistently bigger than that of the measurements, which results in a decreasing friction estimate due to the higher suggested tyre force of the measurement. Also note that for both SDRE and EKF this effect becomes larger once for smaller friction estimates, as the fraction of generated tyre force divided by potential tyre force becomes larger.

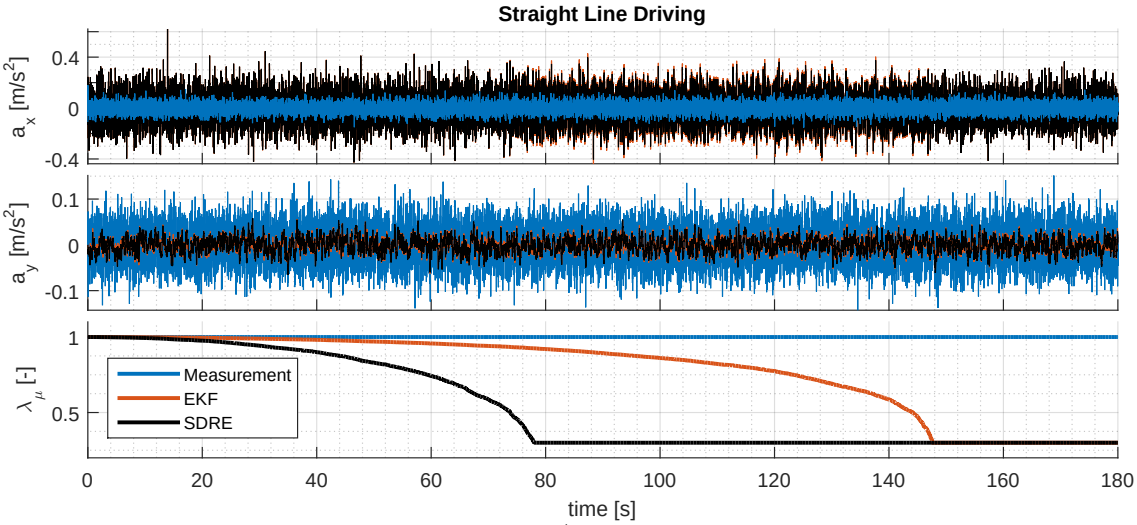


Figure 4.2: Straight line driving of 30m/s, with noise corrupted wheelspeeds signals

Obviously this friction drift is undesirable. It is important to note that if sufficient acceleration would be present, such that the noise band is completely positive or completely negative, this problem is not present. This is used to implement logic that prevents the described problem. The filter is implemented such that only the innovation of acceleration is used if the moving average of the measured output is bigger than a given threshold

$$\frac{1}{n}\sum_{i=0}^{n-1} a_{x,y_{k-i}} > a_{\text{thres}_{x,y}} \tag{4.29}$$

where $a_{\text{thres}_{x,y}}$ represent the selected threshold, n the number of samples used in the moving average filter and $a_{x,y_k}, a_{x,y_{k-1}} \ldots, a_{x,y_{k-i}}$ the sequence of acceleration samples. This method solves the problem effectively.

Additionally, this simple example also shows the potential benefit of modelling the wheel dynamics for the observer. Since including the wheel dynamics in the state estimator results in a filtering effect on the wheelspeed signals. This would reduce the noise present on the wheelspeed inputs, which in turn could significantly improve the quality of the model outputs.

4.3.4 μ Error Covariance Saturation

Due to the absence of dynamics in the λ_μ estimate, the element in matrix $\mathbf{P}$ can, on some occasions, grow indefinitely. When excitation of this augmented state is low, terms in (2.15) result in growing element in $\mathbf{P}$ corresponding to that state. Since the augmented state has no dynamics, terms in $\mathbf{A}$ corresponding to this state are all zero. This results in the term $\mathbf{AP} + \mathbf{PA^T}$ having a zero value on the diagonal element corresponding to the λ_μ state.Additionally, when the state is not sufficiently excited, friction becomes unobservable in the outputs, resulting in the $\mathbf{PH^T V^{-1} HP}$ term also being zero at the element of interest. The process noise covariance

matrix $\mathbf{G}$ is then the only non-zero term remaining. Integrating this term over time leads to an ever growing uncertainty in the λ_μ state.

This could result in problems when the vehicle is not excited in the friction state for an extended period of time. With large uncertainty in the λ_μ state, a small difference in output could already lead to large corrections in the λ_μ state, whenever the element in $\mathbf{H}$ is non-zero for the first time after a long period of non-excitation.

However, one could argue that the true uncertainty in λ_μ is bounded, since friction cannot become negative. Also change in road surface or road conditions hardly ever exceed a factor 2. One could assume therefore an upper bound on the variance of the friction scaling state. With no prior knowledge of road friction, a uniform distribution can be assumed for $\lambda_\mu \in [0, 2]$ to obtain a worst case and conservative upper bound. The variance of such a distribution is given by (4.30), where ub and lb denote the upper and lower bound of the distribution respectively.

$$\sigma^2_{\text{uniform}} = \frac{1}{12}\left(\text{ub} - \text{lb}\right)^2 \tag{4.30}$$

4.4 Performance Criteria

Comparing different estimators is not straightforward as multiple criteria should be considered. This section describes the various criteria that should be considered, when tuning and comparing the various estimators.

4.4.1 Sequence Statistics Validation

Estimator performance can be increased by increasing the accuracy of the model and by tuning of the noise intensities. However, when tuning these noise intensities, the resulting values of error covariance should match with the statistics of the actual errors and innovations. This can be used to test the consistency of the estimation algorithms. As outlined in Chapter 2, both process and measurement noise are assumed to be additive, Gaussian and zero-mean. Typically, when tuning an observer, the true state of the system is not available or measurable. Only the in- and outputs of the system are known. Hence, the error in true state cannot be used to validate the functionality of the observer. However, one could use the innovation since it is available for assessment. Error and innovation are defined respectively as

$$\mathbf{e}(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t) \tag{4.31}$$

$$\boldsymbol{\nu}(t) = \mathbf{z}(t) - \hat{\mathbf{z}}(t) \tag{4.32}$$

The resulting sequence of errors and innovations after running the estimation algorithm for an arbitrary amount of time can be used to evaluate the performance of the estimator. The variances of these differences are summarized by the covariance matrices

$$\mathbf{P} = E\left[\mathbf{e}\mathbf{e}^{\mathbf{T}}\right] \tag{4.33}$$

$$\mathbf{S} = E\left[\boldsymbol{\nu}\boldsymbol{\nu}^{\boldsymbol{T}}\right] \tag{4.34}$$

The error covariance matrix $\mathbf{P}$ describes the uncertainty in each state estimate. In a similar matter, the covariance of the innovation $\mathbf{S}$ describes the uncertainty in output. This can be found by considering the uncertainties in the state estimates in combination with the additive noise. Propagation of the error covariance propagation was already discusses in Chapter 2. The innovation covariance approximation is found by

$$\mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^{\mathbf{T}} + \mathbf{V} \tag{4.35}$$

The diagonal elements in the covariance matrices describe the variance in the errors and innovation of each state and output. These values can thus be used to provide bounds on the expected error and innovation, as demonstrated in [12, 17]. These bounds can be used to evaluate the performance and consistency of the estimator. Under the assumption of Gaussian noise, the 95% confidence intervals are given by

$$-2\sqrt{P_{ii}} < e_i < 2\sqrt{P_{ii}} \tag{4.36}$$

$$-2\sqrt{S_{ii}} < \nu_i < 2\sqrt{S_{ii}} \tag{4.37}$$

These bounds can be used to evaluate if the errors and innovations of an evaluated dataset are represented well by the error and innovation covariance matrices. This can be used to validate the choices for noise intensities of process and measurement noise.

4.4.2 Convergence Speed

Aside from the statistics concerning the innovation and estimation error, it is also of interest, to what extend the estimation algorithms are capable of converging from a wrong estimate towards the correct value. Naturally, also the time it takes for the estimate to converge can be used as a measure for performance.

4.4.3 Filter Parameter Robustness

Although stability of the EKF and SDRE estimator shall neither be proven rigorously due to the difficulties outlined in Chapter 1, an assessment of the estimators robustness under estimator parameters can be performed. Specifically parameters that can change for a given vehicle are of interest. This could include changing mass parameters corresponding to changing load and passengers.

4.4.4 Computational Effort

Since the estimator is intended for real time implementation, the required computational effort is also of interest. The hardware in commercial vehicles often has only limited computational power, or limited capacity reserved for the vehicle state estimator. Therefore it is beneficial to have an estimator that has a lower computational load.

4.5 Summary

This chapter describes the relevant equations for implementation of the algorithms presented in Chapter 2 for usage in the vehicle state estimator. In particular, the process of derivation of the Jacobian matrices used in the EKF estimator are presented. In addition, the state dependent coefficients used in the SDRE estimator are also presented. They are derived using a heuristic approach, in which an attempt has been made to mimic some of the behaviour present in the EKF estimator.

Additional logic that is implemented is also presented. This logic does not increase the effectiveness of either of the algorithms, but rather increases estimator performance in general. Finally, some considerations are presented which can be used to evaluate the performance of the estimators.

Chapter 5

Simulation Results and Analysis

This chapter will describe the testing of the estimator using model generated datasets instead of actual vehicle measurements. This allows to evaluate the performance of tracking states that are not available in the real vehicle, such as friction. Robustness of the filters against parameter changes is also studied. Additionally, conditions that can not be tested on the experimental set-up (asymmetric braking) will be tested via these simulations. Finally, some suggestions are made to utilize the statistics of the innovation in order to improve the convergence rate for changing system parameters.

5.1 Plant

To generate the required datasets, a plant model is used. This ensures that true states are accessible without measurement noise. The dynamics of the model are described with the same equations as described in Chapter 3. However, they are extended by including the wheel dynamics. Process noise is also included to create a discrepancy between the model and observer models, and measurement noise is added to more realistically represent real measurements. All simulations in this chapter are performed at a sample-rate of 100 Hz.

Wheel dynamics are included by applying an external braking or driving moment to the wheels. Here I_p is the polar moment of inertia of the wheel, R_{l_i} the loaded radius of the wheel and M_{ext_i} the externally applied braking or driving moment.

$$\dot{\omega}_i I_p = M_{\text{ext}_i} - R_{l_i} F_{x_i} \tag{5.1}$$

5.2 Friction Estimation

This section will discuss the ability of estimating the augmented friction state. It will discuss tracking of the friction estimate for steady-state cornering. Figure 5.1 shows a time response of constant cornering where friction is initially dropped, after which friction is increased again. It can be seen that the performance of the filters is drastically different.

The response of the estimators in steady-state cornering conditions are used to study the convergence rate as a function of the lateral excitation. The vehicle is initialised at a fixed forward velocity. A fixed steering angle is then applied at $t = 0.5s$ and maintained throughout the simulation. An array of steering wheel angles is used, $\delta_{sw} = [0, 5, 10, \ldots, 155, 160]$. A small driving torque is applied to maintain a constant forward velocity. A steady-state lateral acceleration is approximately achieved at $t = 1.5s$. At $t = 10s$ the friction is reduced from 1 to 0.5, or increased from 0.5 to 1. The settling time of the friction estimate is then compared for the EKF and SDRE observers. An overview of the results can be seen in Figure 5.2, which shows

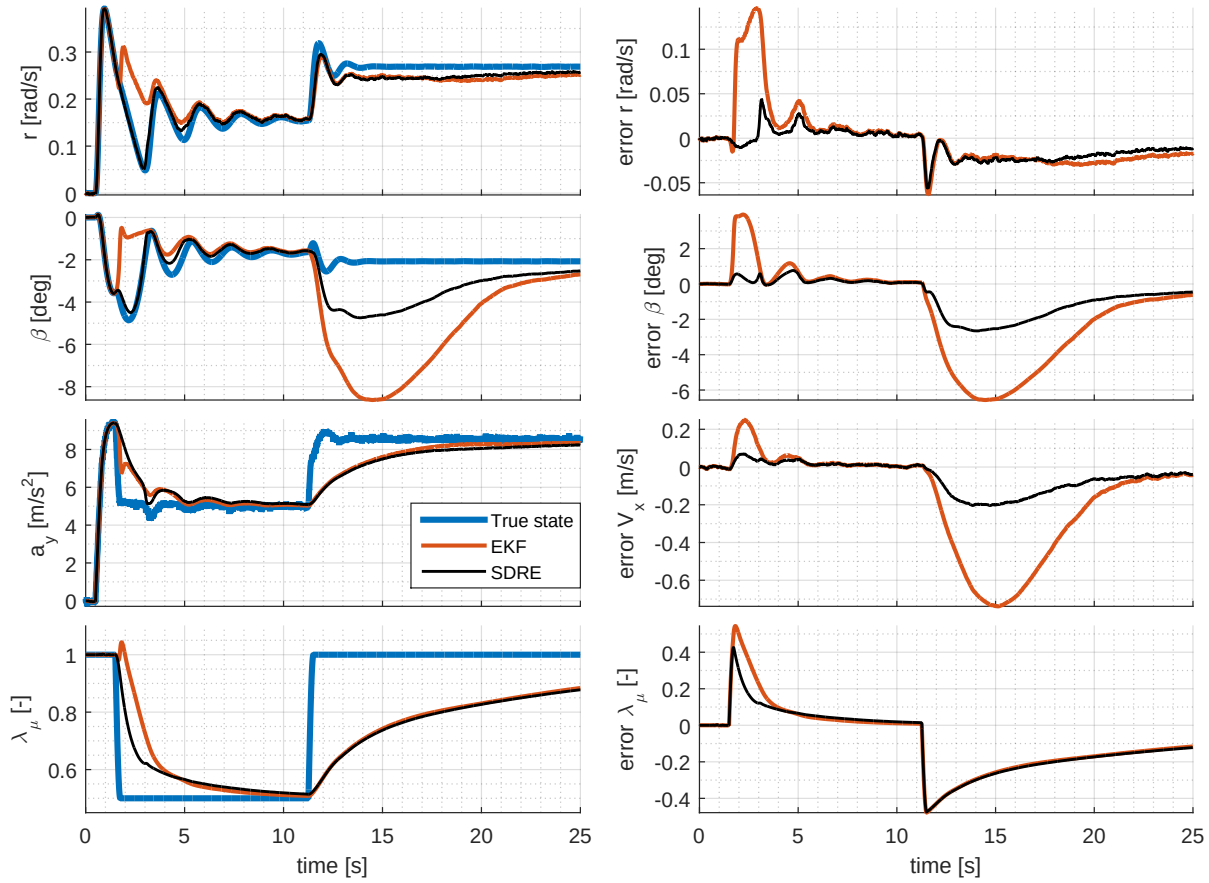


Figure 5.1: Estimation results for double friction change in steady-state cornering conditions at $V_x = 30m/s$ with $\delta_{sw} = 90[deg]$

the time needed for each estimator to decrease the error to below respectively 5% and 36.8% of the step in friction $|\Delta\lambda_\mu| = 0.5$. These values correspond to the settling time and time constant respectively. It should be noted that the convergence rate heavily depends on the selected filter settings related to the error covariance. The influence of initial condition is minimal, as the initial estimate is chosen identical to the true state. Additionally, the step in friction was applied after $\mathbf{P}$ is converged, such that effect of initial condition $\mathbf{P_0}$ is minimized. Significantly faster convergence rates can be achieved with different noise settings. However, for the results presented here, identical values are selected to those that worked well in the experiments.

Figure 5.2 shows that a minimum amount of excitation is clearly needed for convergence, as can be seen in the upper two plots. Additionally, It can be seen that for bigger lateral accelerations, the time required for convergence is lower. However, the upper right plot shows that for further increasing lateral acceleration, the convergence seems to deteriorate. However, this is not the complete picture. The plots from the figure show only the time required to decrease the error below two thresholds. This means that the initial response speed is not truly considered. Something similar can also be seen in Figure 5.1, where it can be seen that the initial convergence rate of the SDRE estimator is higher. However, the EKF estimator requires less time to decrease the friction error bellow the 5% error threshold. To consider not only time required for convergence, but also include the initial response, a second criteria is used to penalise larger errors more severely. To this extend, the integral square error is used

$$||e_{\lambda_\mu}(t)||_2 = \sqrt{\int_0^{t_{\text{err}}} (\lambda_\mu(t) - \hat{\lambda}_\mu(t))^2 \, \mathrm{d}t}. \tag{5.2}$$

This time-average estimation error is given for $t_{\text{err}} = 10s$ in Figure 5.3. In addition to friction the figure also gives the time-average estimation error of the side slip angle. It should be noted that since the true slip angles also increase for increasing lateral accelerations, an increase in error is always to be expected. Therefore, the integral square error for the body slip angle is taken from the normalised signal. It can be seen that both EKF and SDRE show fairly similar integral square errors. However, the lower left plot of Figure 5.3 shows significant better tracking for SDRE, as the body slip angle integral error is significantly smaller.

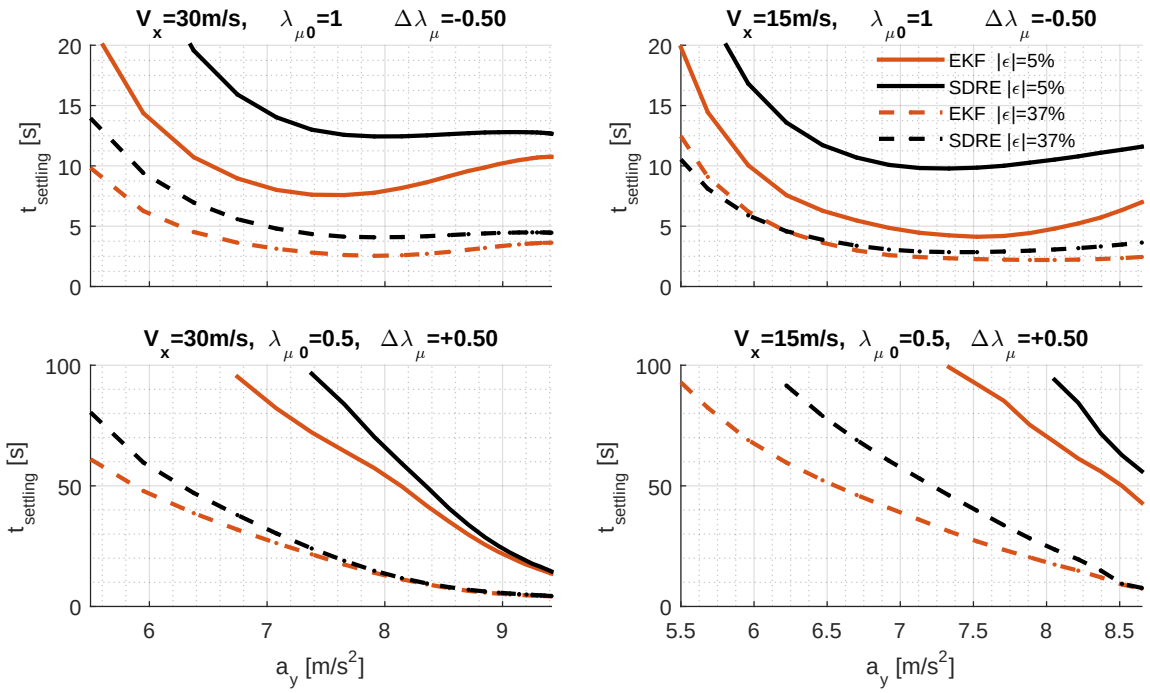


Figure 5.2: Convergence time vs. lateral acceleration for EKF and SDRE during steady-state cornering. Settling time for bounds of $\epsilon = 0.05|\Delta\lambda_\mu|$ and $\epsilon = 0.386|\Delta\lambda_\mu|$

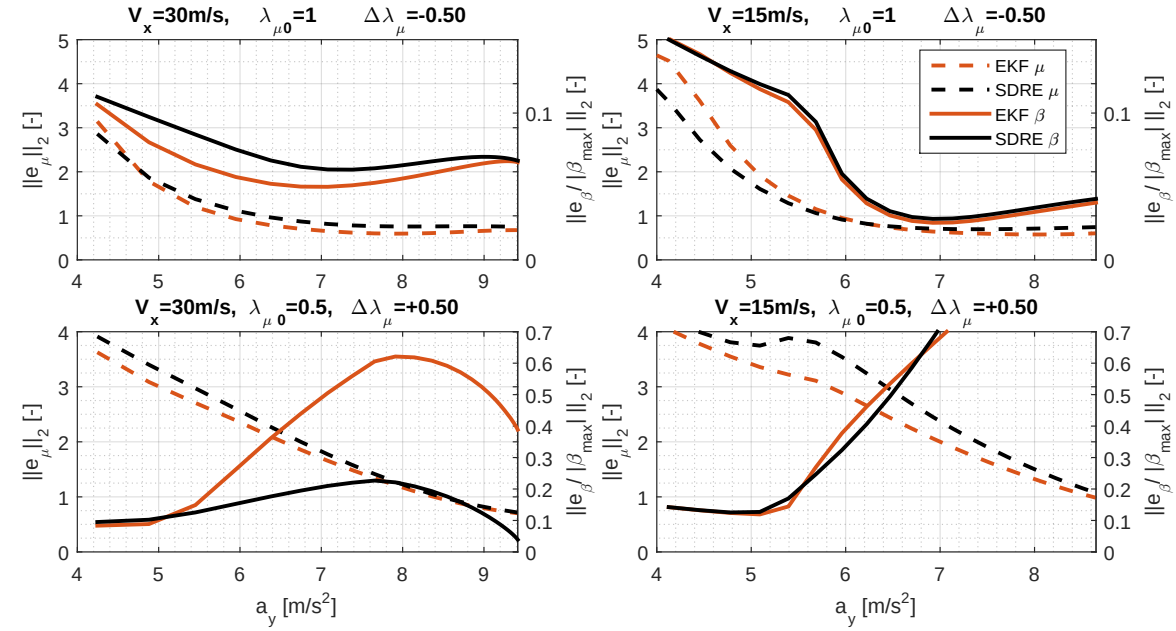


Figure 5.3: Integral Square error for λ_μ (left axis) and β (right axis)

5.3 Parameter Robustness

So far plant and model are identical with exception of process noise. However, in real world applications the model is hardly ever this accurate. Therefore the influence of parameter changes is studied. For vehicle state estimation, especially the tyre characteristics and vehicle mass are not always fully known. This is due to the identification measurements often being absent for commercial tyres. Additionally, the load of the vehicle can vary (e.g. the amount of passengers). A sine with dwell manoeuvre will be used to compare the robustness of the various filters. This manoeuvre is intended to excite the system in such a way that more extreme slip angles are encountered. A forward velocity of 20m/s and a sine with dwell steering input with maximum amplitude of approximately 22 degrees at the front axle, are used.

5.3.1 Tyre Parameter Uncertainty

Due to the absence of large side slip measurements for the fitting process described in Section 3.3, the behaviour at larger slips is still somewhat uncertain. Therefore, the behaviour of the estimators with a discrepancy in tyre model between model and plant will be discussed. For the simulation the shape factor $C_{x,y}$ of the Magic Formula for the plant has been set to 1.3 (presented as example value in [28]), whereas the model retains the value of 1. Results of such a manoeuvre can be seen in Figure 5.4.

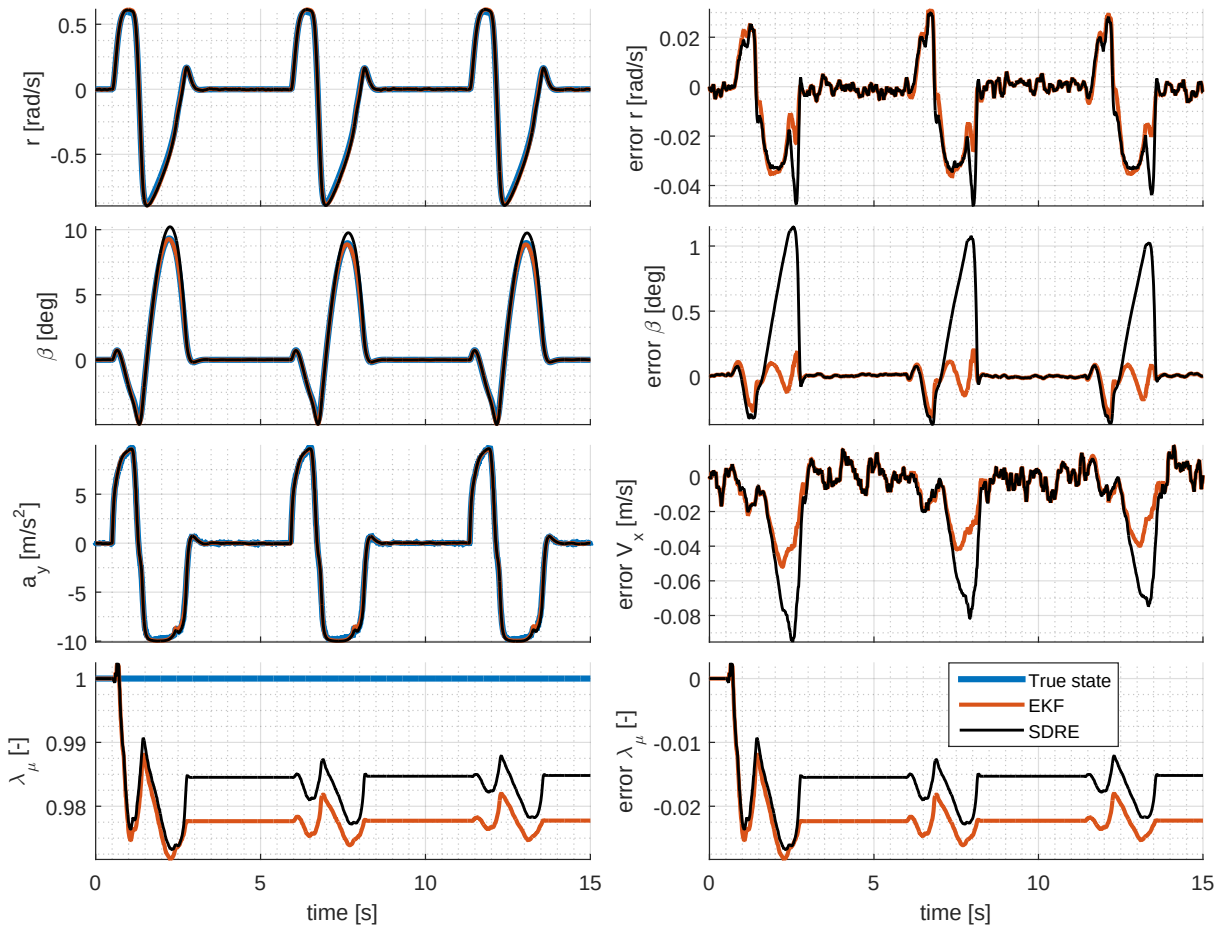


Figure 5.4: 20m/s sine with dwell manoeuvre, plant $C_y = 1.3$ $C_x = 1.3$, observer model $C_y = 1.0$ $C_x = 1.0$

Although a fairly large error in side slip angle can be seen, the filter performs reasonably well, with only minor changes in the friction estimate. However, increase the discrepancy between

model and observer model even further results in deteriorating estimates, especially for the SDRE estimator.

5.3.2 Mass Parameter Uncertainty

A similar experiment is performed with the vehicle mass. To check the robustness of the algorithms for the mass parameter, 240 kg is added to the plant model, of which 180 kg is added to the rear seat, and 60 kg to the trunk of the car. The change in I_{zz} and center of gravity location are also taken into account. It can be seen in Figure 5.5 that the shape of the response is very similar to that of the estimator. The maximum side slip error is approximately 2 degrees for the SDRE estimator and 1 degree for the EKF estimator.

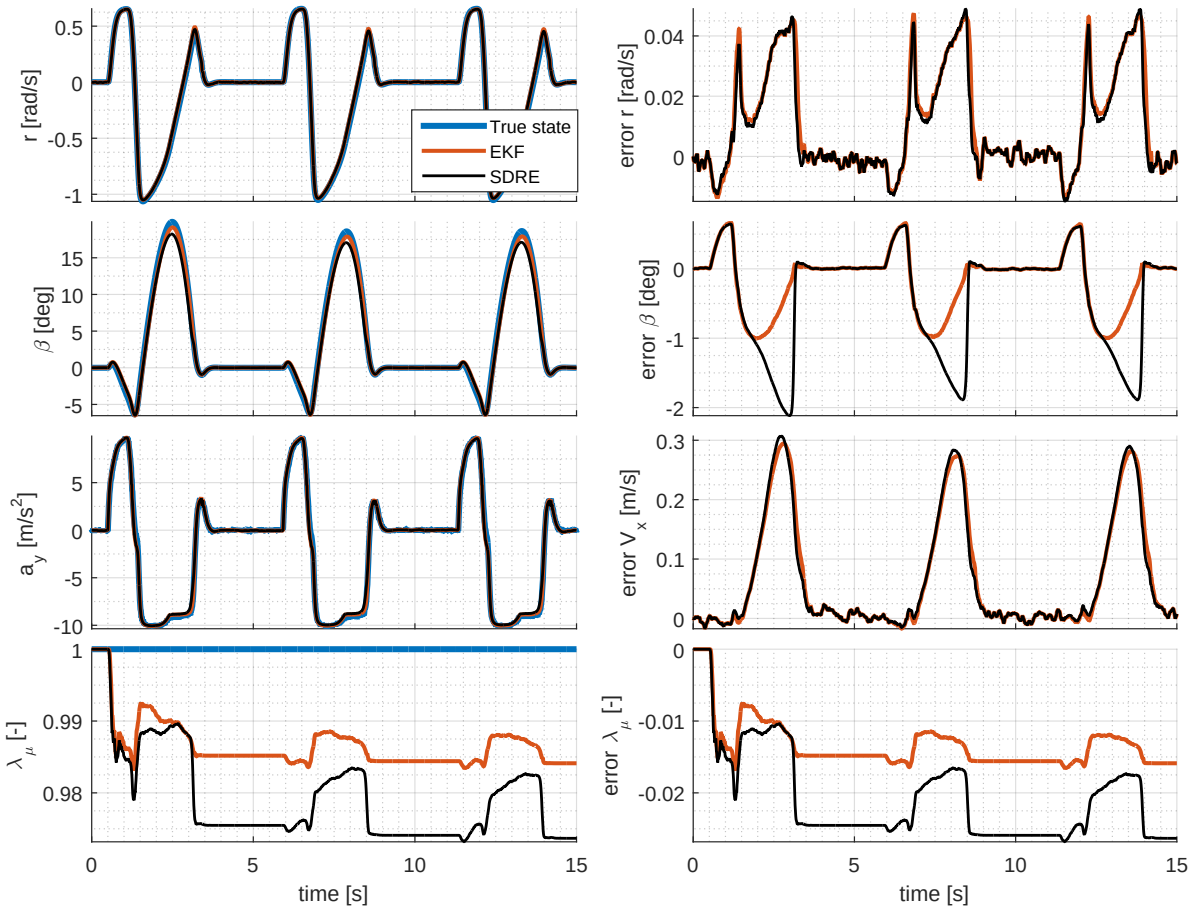


Figure 5.5: 20m/s sine with dwell manoeuvre, plant is 240 kg heavier, with adapted I_{zz} and COG location with respect to observer model

5.3.3 Robustness Filter Tuning

In the previous two examples the process noise matrix $\mathbf{G}$ and measurement noise $\mathbf{V}$ are chosen again identical to that used in the experimental set-up. These values are found using the procedure described in Appendix C. The resulting estimator is fairly robust for inertial and tyre shape factor parameters changes as can be seen from previous figures. If however, the noise intensities are chosen differently the resulting estimators can be significantly less robust to such changes in the plant. If for instance the process noise matrix $\mathbf{G}$ and measurement noise $\mathbf{V}$ are chosen to be very optimistic, such that the process noise is chosen low, the corrections are much smaller and estimation errors can become large. This can be seen in Figure 5.6, where $\mathbf{G}$ is chosen such that the entries corresponding to the dynamics states $\{r, V_y, V_x\}$ are a factor

10 lower and the entry corresponding to the friction a factor 10 higher. This implementation follows from having a high level of trust in the vehicle model, and increasing uncertainty in unknown parameter friction.

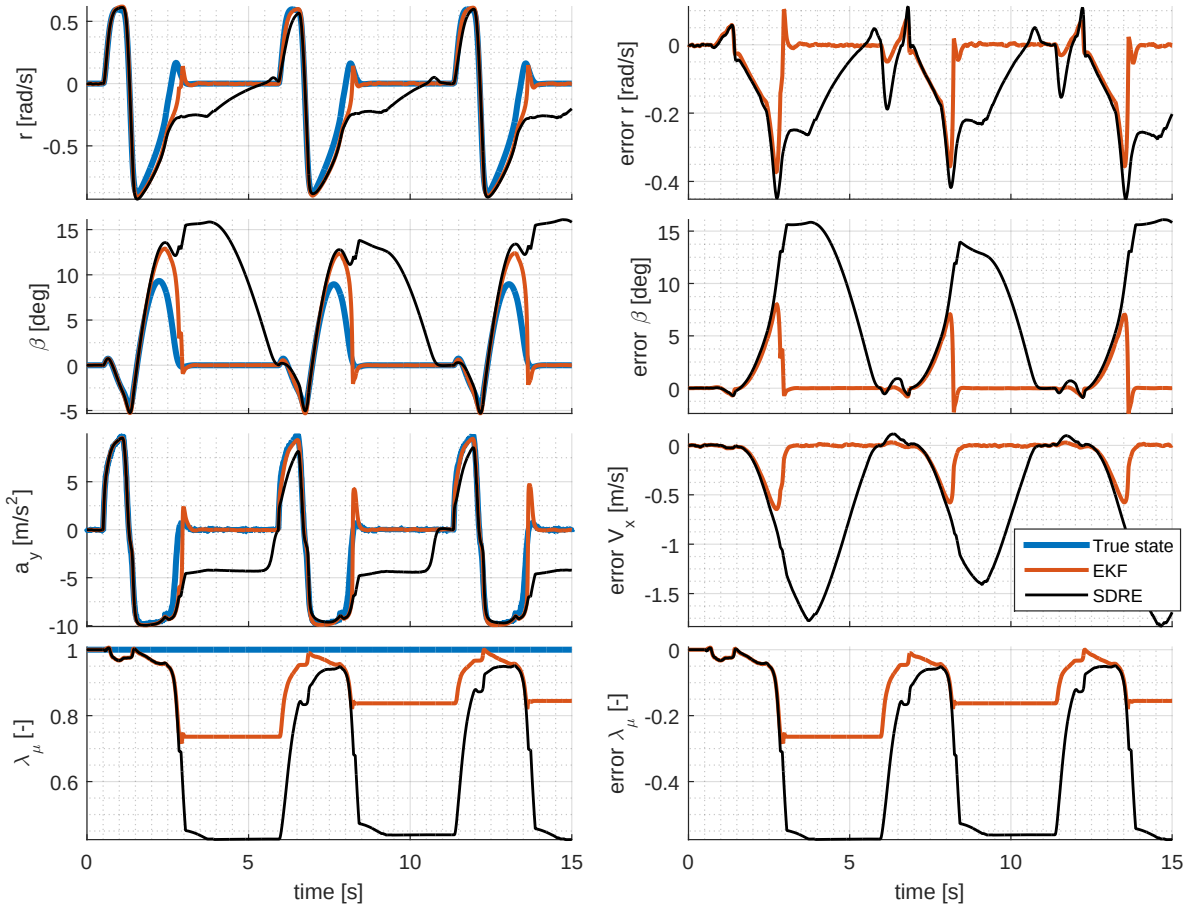


Figure 5.6: 20m/s sine with dwell manoeuvre, plant $C_y = 1.3$ $C_x = 1.3$, observer model $C_y = 1.0$ $C_x = 1.0$, Very optimistic $\mathbf{G}$ and $\mathbf{V}$

Clearly the states are no longer correctly tracked. The SDRE estimator significantly deviates from the true trajectory. Note that the EKF estimator also results in large side slip errors of 5 degrees. The combination of the smaller measurement noise $\mathbf{V}$ and smaller process noise $\mathbf{G}$ results in smaller values for the error covariance matrix $\mathbf{P}$. This in turn results in much smaller state corrections which allows the state estimates to deviate from the true trajectory.

5.4 Simultaneous Steering and Asymmetric Braking

Ultimately, the objective of the vehicle state estimator is to track the vehicle states for collision avoidance systems. Systems that allow the car to simultaneously actuate individual brakes and steering wheel have proven to be superior for this goal. The Prius that is used for the experiments is not capable of actuating the brake asymmetrically. Hence, this functionality can not be tested on the experimental set-up. Instead, simulations are used to verify that the state estimator works well in these conditions. A set of control inputs that is generated using the SDRE based controller presented in [3]. This SDRE based controller assumes an all-wheel drive vehicle, such that wheel torques can be controlled in both directions. This has no direct impact on the functionality of the estimator, other than that the rear wheels no longer provide a accurate reference velocity. Therefore, for this particular simulation the GPS speed (4.3) is used to replace the velocity measurements from the undriven wheels (4.4). To create larger differences between observer and

model, both the model differences of mass and tyre parameter that were illustrated earlier are used. The results can be found in Figure 5.7. It can be seen that despite fairly large differences between model and plant, the states can still be tracked relatively accurately.

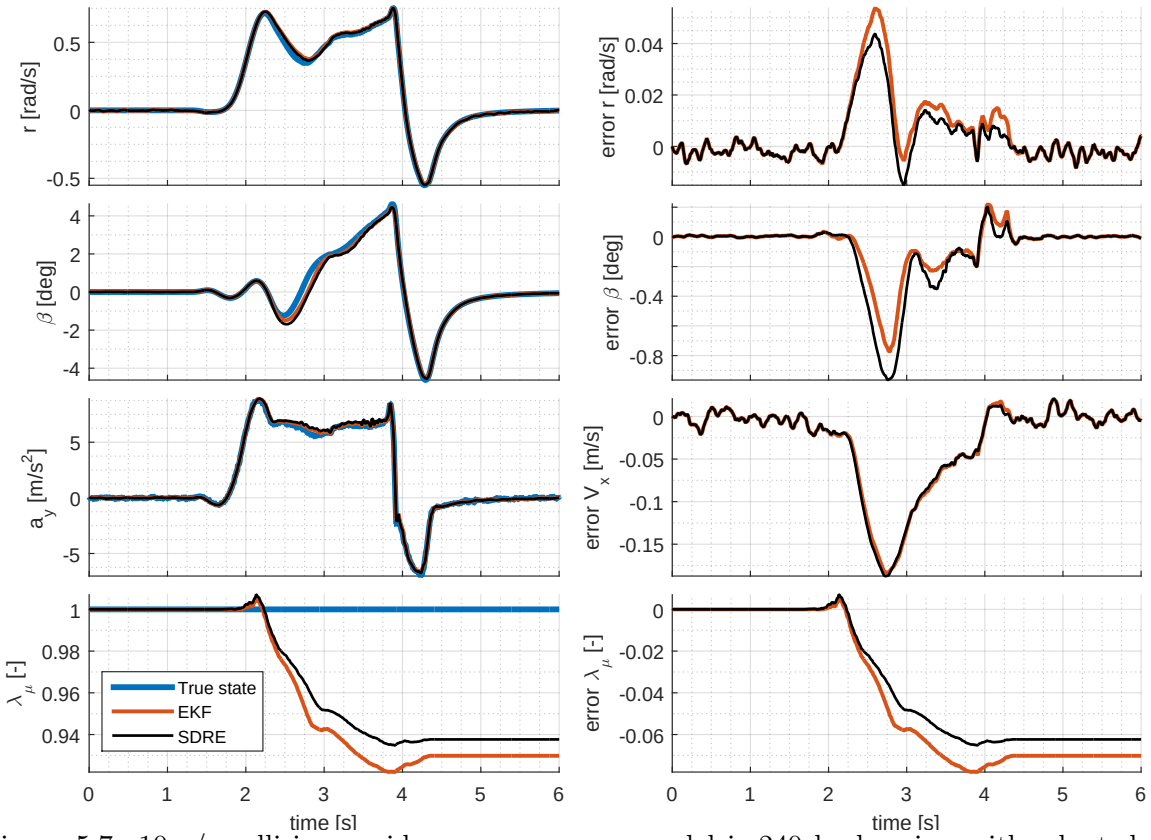


Figure 5.7: 10m/s collision avoidance manoeuvre, model is 240 kg heavier, with adapted I_{zz} and COG location with respect to observer model, model $C_y = 1.3$ $C_x = 1.3$, observer model $C_y = 1.0$ $C_x = 1.0$

5.5 Error and Innovation Statistics

As briefly mentioned in Section 4.4, statistics involving the innovation can be used to evaluate the performance of the estimator. It will be demonstrated here that the selected noise intensities for the filter can be verified to accurately represent the true noise, as is described in [17]. Additionally, it will be shown that non-modelled dynamics, such as a drop in friction, can be recognised. If the vehicle model is accurate enough, this can be used to temporarily increase the correction gain for the friction correction.

5.5.1 Noise Intensity

First the case with complete model knowlegde is considered. Here the model is exact and noise intensities are also known perfectly. Due to the strong correlation of the wheelspeed signals with the other states, the noise on input wheelspeeds is set to 0. The noise illustrated with respect to the V_x signal is taken from the GPS sensor. The results of the innovation sequence for both algorithms are shown in Figure 5.8. In the figure the innovation sequence and a moving average are shown in respectively blue and purple. Two bounds are also given representing the 2σ gates for only measurement noise, and measurement noise plus uncertainty in states. The innovation sequence lies within the 2σ-gate for 95% of the samples. This suggest that the selected noise is in fact a good representation of the true noise.

Next the case where true process noise variance is a factor 1000 higher than is used in the filter is considered , which is shown in Figure 5.9. It becomes clear that the innovation sequences become less 'white', and some time correlation can be detected.

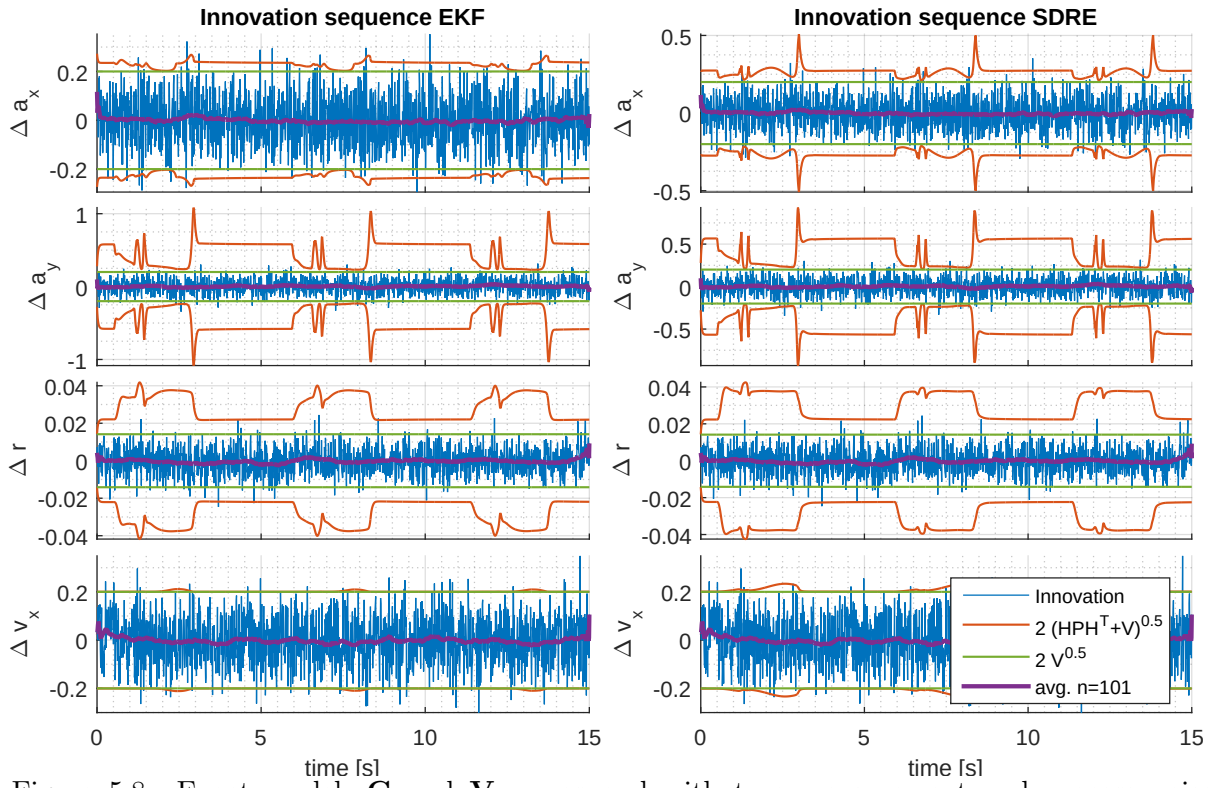


Figure 5.8: Exact model, $\mathbf{G}$ and $\mathbf{V}$ correspond with true measurement and process noise intensities, repeating sine-with-dwell manoeuvre at 20m/s

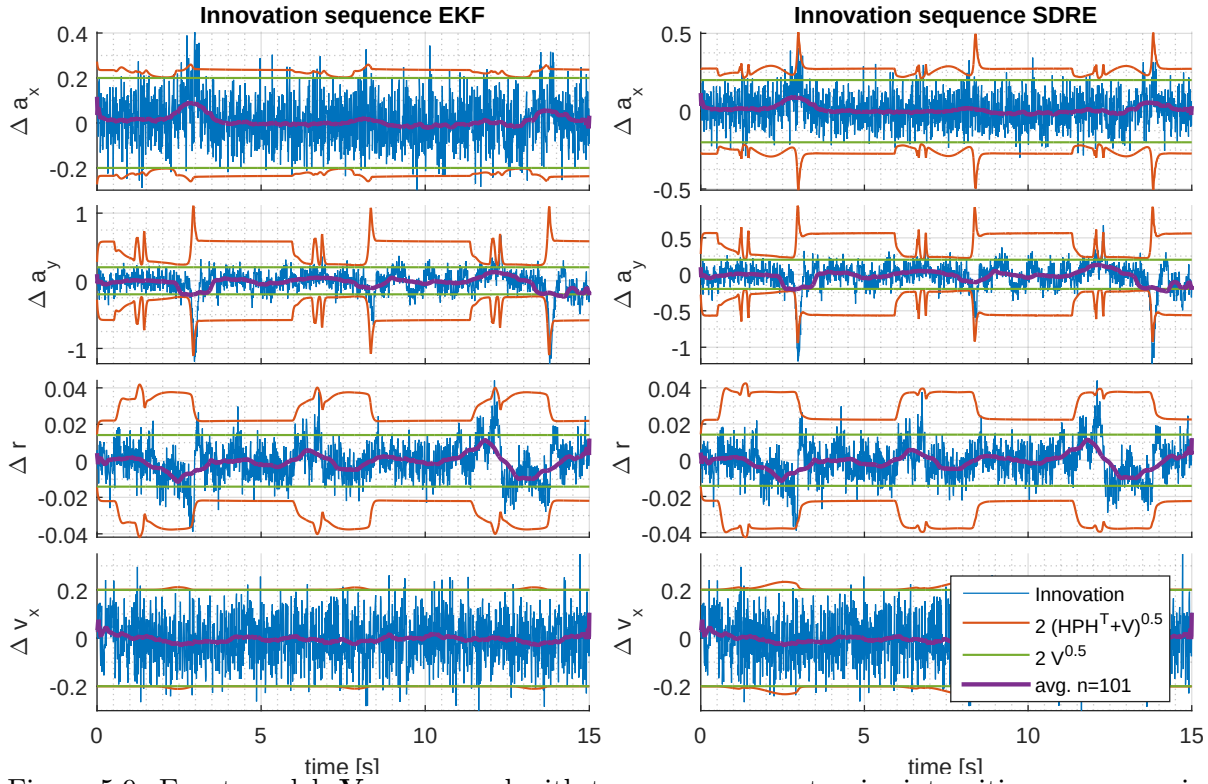


Figure 5.9: Exact model, $\mathbf{V}$ correspond with true measurement noise intensities, process noise variance underestimated by factor 1000 in $\mathbf{G}$, Repeating sine with dwell manoeuvre at 20m/s

5.5.2 Parameter Changes

Finally, and perhaps most interestingly, the case is considered where a change is applied to the plant, which is not modelled. In this case a drop in friction as is also used in Section 5.2 will be presented. Again at $t = 2.5$ a drop in true friction is applied. Clearly, the innovations are no longer white and exceed the 2 σ bounds in multiple outputs. This is illustrated in Figure 5.10

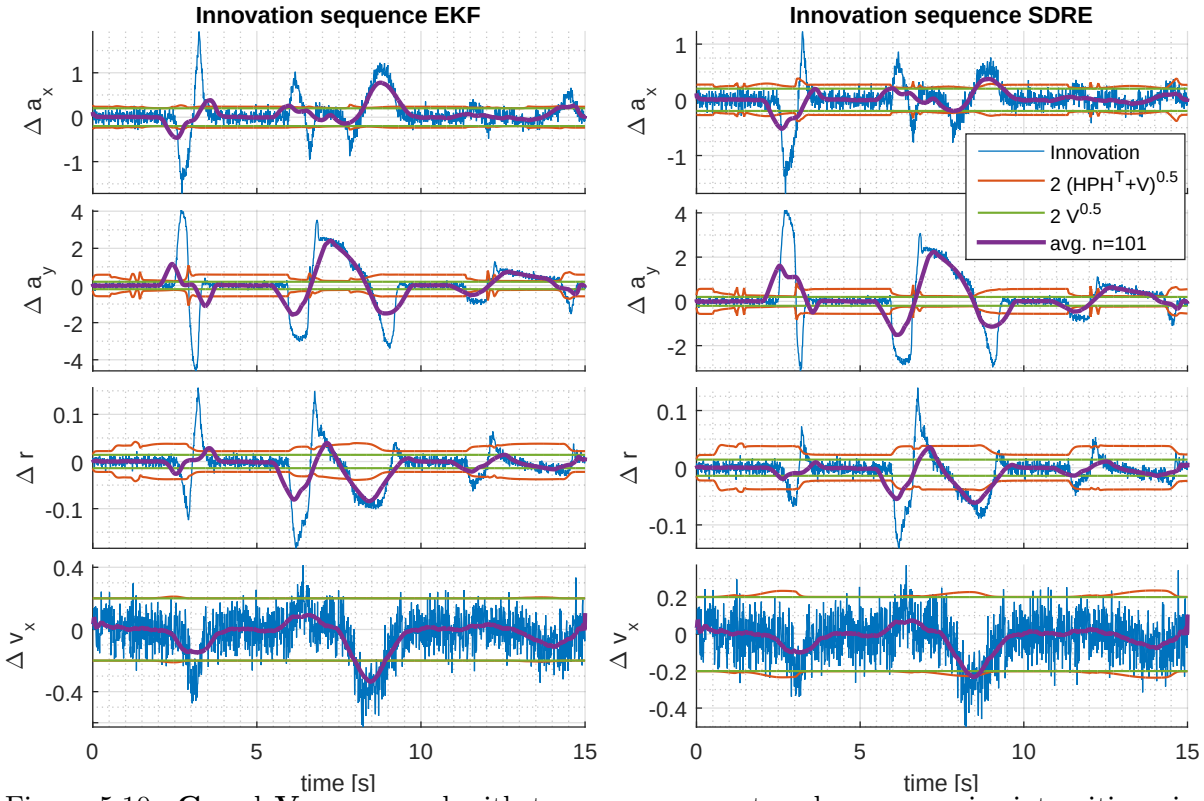


Figure 5.10: $\mathbf{G}$ and $\mathbf{V}$ correspond with true measurement and process noise intensities, sine width dwell manoeuvre, drop in friction

This behaviour can be utilized to amplify the correction in friction, since if the model is accurate, the big discrepancy must be to a change in parameter. Effectively this can be used to penalise large errors more, such that in wet conditions the friction drop is detected earlier. However, it should be noted that this requires the model to be very accurate. Any unmodelled dynamics or other inconsistencies in the model can results in similar behaviour, in which case it is undesired to amplify the correction in friction. A wrong mass parameter for example leads to similar inconsistencies as can be seen in Figure 5.11. Therefore such logic should be implemented with care and sufficiently large thresholds should be chosen. A first simple attempt has been made to implement this conceptual logic. The sine with dwell manoeuvre is again used in combination with a heavier plant than used in the model. The logic is implemented as following

$$\mathbf{K} = \begin{cases} diag\{\begin{bmatrix}1 & 1 & 1 & K_\mu\end{bmatrix}\}\mathbf{P}\mathbf{H}^\mathbf{T}\mathbf{V}^{-\mathbf{1}} & \text{if } |\hat{a}_x - a_x| \leq \nu_{a_{x_{\text{thres}}}} \vee |\hat{a}_y - a_y| \leq \nu_{a_{y_{\text{thres}}}} \\ \mathbf{P}\mathbf{H}^\mathbf{T}\mathbf{V}^{-\mathbf{1}} & \text{if } |\hat{a}_x - a_x| > \nu_{a_{x_{\text{thres}}}} \vee |\hat{a}_y - a_y| > \nu_{a_{y_{\text{thres}}}} \end{cases} \tag{5.3}$$

Where K_μ is the gain factor with which the friction correction is amplified and $\nu_{a_{x_{\text{thres}}}}$ and $\nu_{a_{x_{\text{thres}}}}$ respectively the thresholds in longitudinal and lateral direction. $K_\mu = 4$, $\nu_{a_{x_{\text{thres}}}} = 4$ and $\nu_{a_{y_{\text{thres}}}} = 1$ are selected for this example. The results can be seen in Figure 5.12. A comparison is made between the friction convergence for the normal estimators and estimators that utilize this conceptual logic. It can be seen that the friction converges much faster in the case of the implemented logic, and therefore also results in better side slip angle estimates.

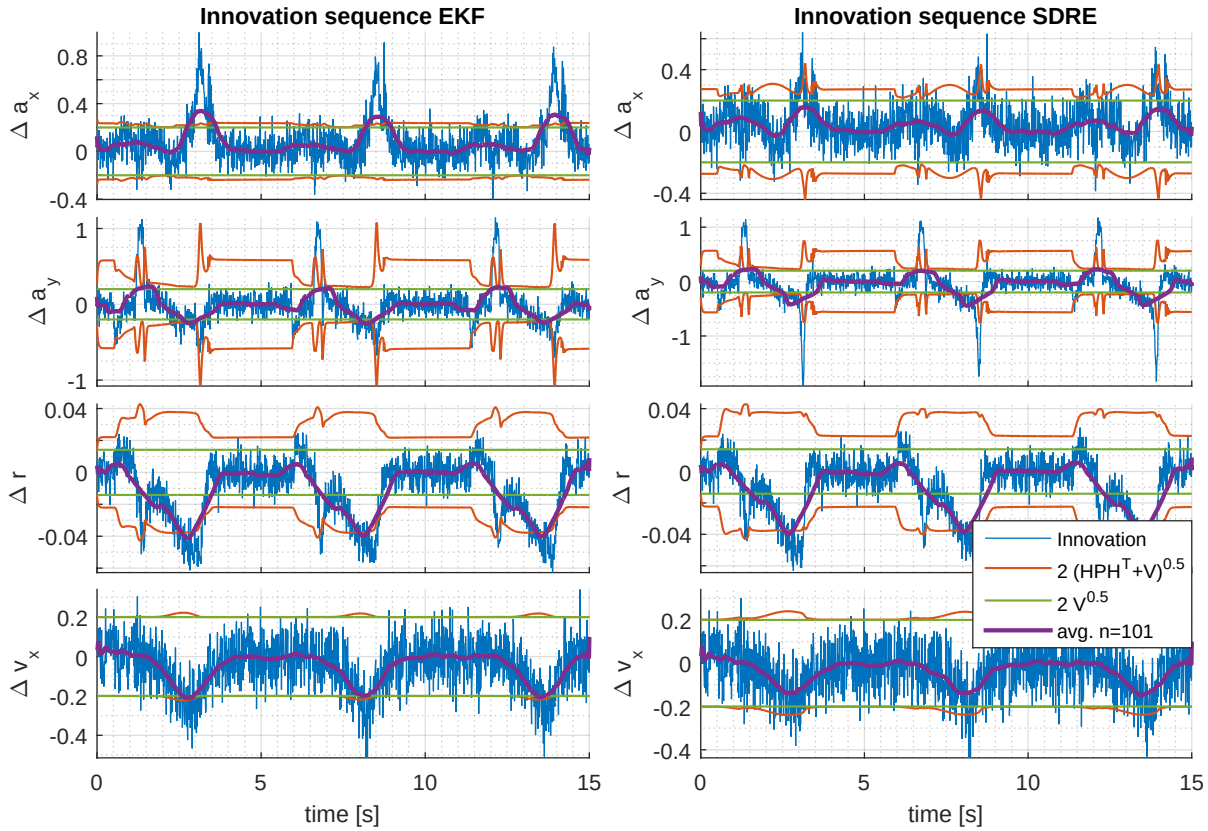


Figure 5.11: Mass of model is 240kg heavier than observer model, $\mathbf{G}$ and $\mathbf{V}$ correspond with true measurement and process noise intensities, sine width dwell manoeuvre

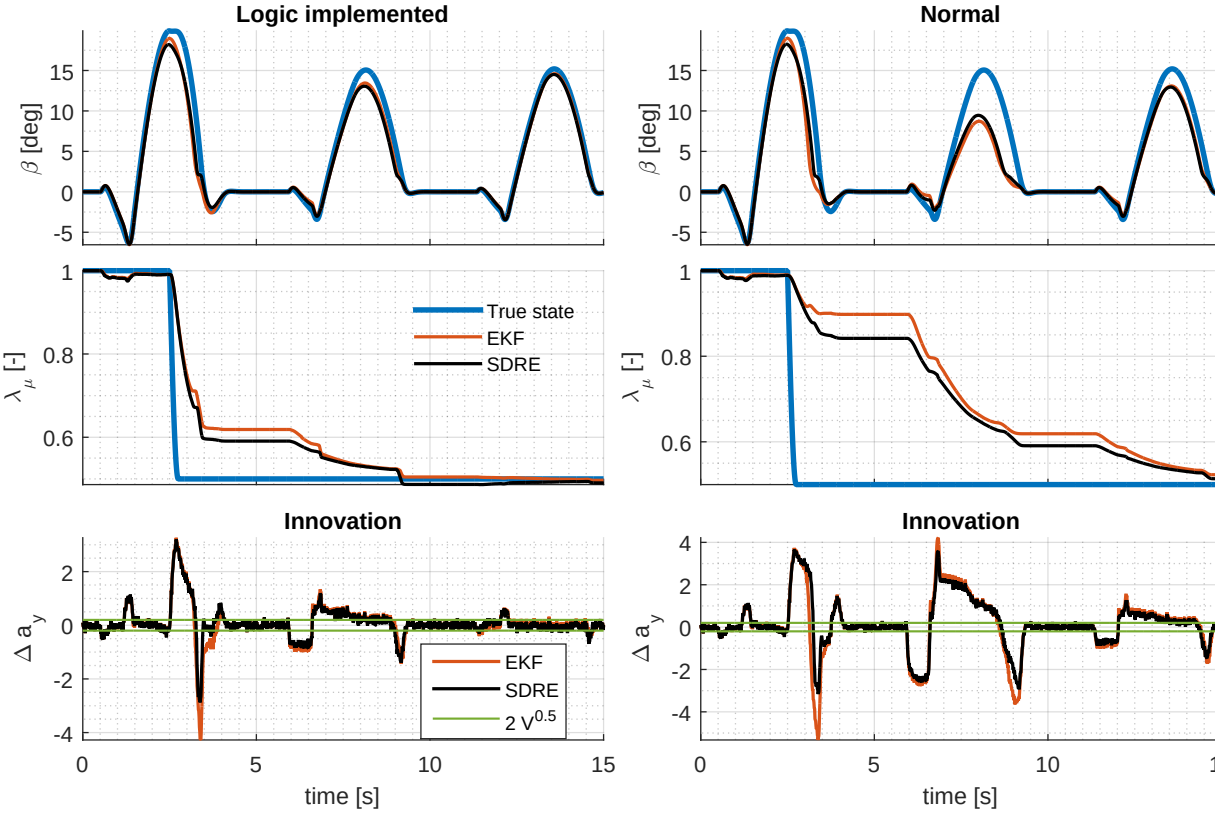


Figure 5.12: Results for suggested logic, as given in (5.3). Mass of model is 240kg heavier than observer model, $\mathbf{G}$ and $\mathbf{V}$ correspond with true measurement and process noise intensities, sine width dwell manoeuvre

With the described logic implemented, friction estimation is significantly accelerated. The simulation described in Section 5.2 is repeated here, with the proposed settings. The results can be seen in Figure 5.13 and Figure 5.14.

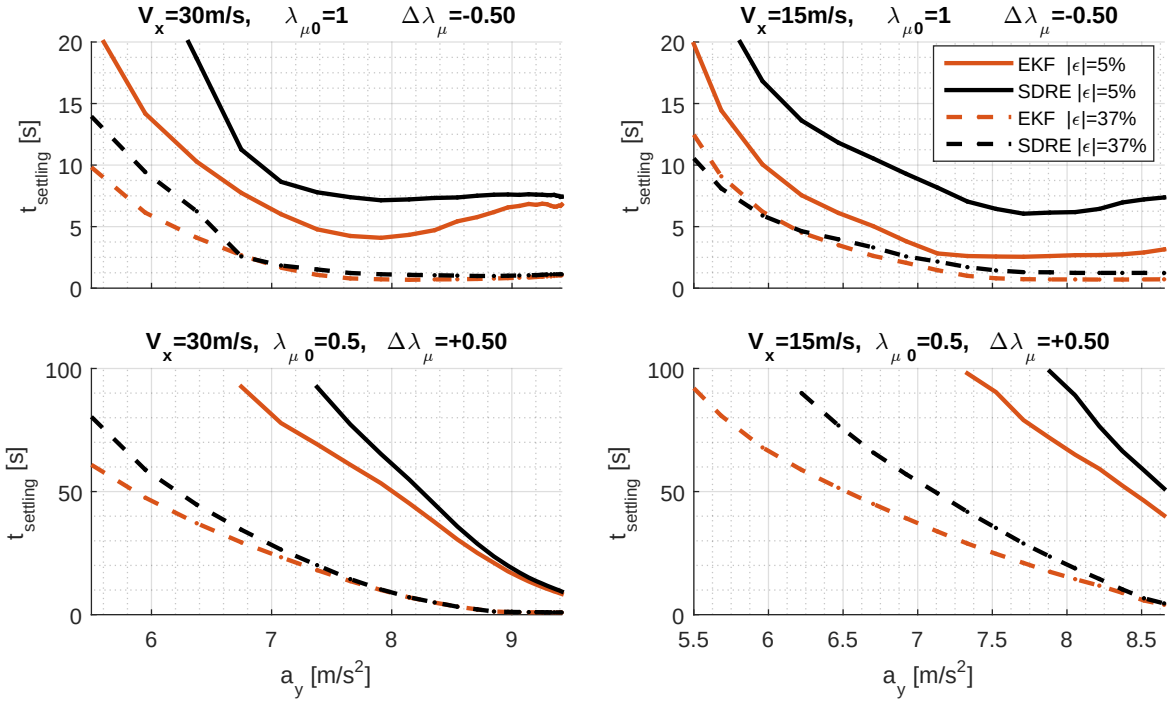


Figure 5.13: Convergence time vs. lateral acceleration for EKF and SDRE during steady-state cornering. Settling time for bounds of $\epsilon = 0.05|\Delta\lambda_\mu|$ and $\epsilon = 0.386|\Delta\lambda_\mu|$, Results are illustrated for implemented logic $K_\mu = 4$, $\nu_{a_{x_{\text{thres}}}} = 4$ and $\nu_{a_{y_{\text{thres}}}} = 1$

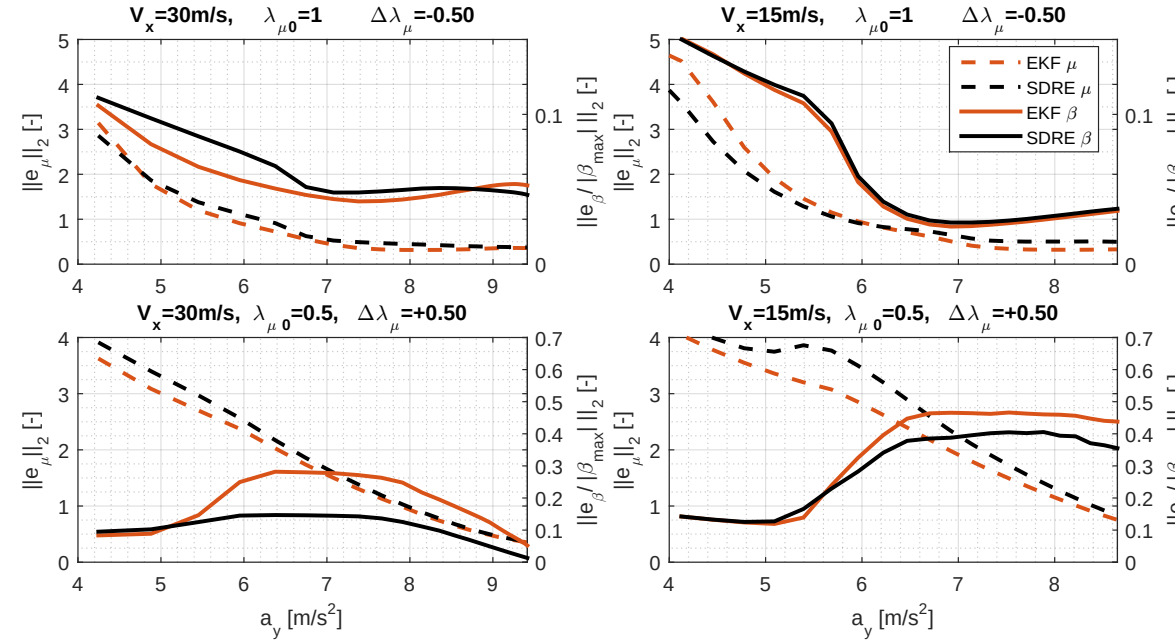


Figure 5.14: Integral Square error for λ_μ (left axis) and β (right axis), Results are illustrated for implemented logic $K_\mu = 4$, $\nu_{a_{x_{\text{thres}}}} = 4$ and $\nu_{a_{y_{\text{thres}}}} = 1$

5.6 Computational Load

In software intended for real-time applications, such as the vehicle state estimator, it is important to consider the computational load of the algorithm. Only a limited amount of the hardware's computational power can be allocated for the observer, making it an important resource to consider. A quick comparison between the EKF algorithm and SDRE algorithm shows a significant difference in computational load. For 3000 seconds of simulated time, the computational time required is approximately 340 and 43 seconds for respectively the EKF and SDRE estimator. It is worth noting that for increasingly complex tyre models, the computation time of the Extended Kalman Filter increases drastically (up to factor 3). This is due to the significantly more complex expressions for the Jacobian matrices.

5.7 Summary

This section will summarize the findings of the tests that are performed to test the two observers. An extension to the plant model is discussed to simulate the wheeldynamics that provide the inputs to the observers.

Secondly friction estimation is discussed in the described simulation environment. This is done since the reference friction is not directly measurable and hence unavailable in the experimental set-up. It is demonstrated that the implemented SDC form presented in 4.2.2 is capable of tracking the states friction with comparable results to that of the EKF. Both EKF and SDRE are shown to only track changes in friction well from a lower bound of roughly $a_y \approx 6m/s^2$.

Robustness of the filter against wrongly selected vehicle parameters is discussed briefly for the cases of different vehicle mass and different tyre parameters. In both cases the estimators where still capable of tracking the states, but only if the process noise and measurement matrix are selected appropriately. This is tested for challenging manoeuvres with significant body slip angles. Also combined slip with asymmetric brake inputs is used and demonstrated to work properly.

Finally, statistics regarding the innovation sequence are shown for three cases. It is discussed that changes in parameters can be detected and anticipated, if the model is accurate enough. If this is the case, the innovation sequence will be approximately white and within the expected bounds. If a change in parameters then occurs, the innovation will exceed these bounds for multiple consecutive samples, which could be used to trigger an amplified correction in the parameters. A conceptual implementation is presented and demonstrated to significantly increase the speed of convergence for the estimators.

Chapter 6

Experimental Results

This chapter will discuss the real-world experiments in which the Vehicle State Estimator is tested. The model described in Chapter 3 is already shown to be accurate. However, the fitting procedure is done with measurements performed on a dry track. Specific attention will therefore be given to data-set acquired on wet track conditions. Additionally, other limit handling cases, in which the vehicle is operated near the friction limit, will be presented. The performance of the estimator will be discussed for these more challenging conditions.

6.1 Test Set-up

The test set-up used for the experiments is the 2012 Toyota Prius as seen in Figure 6.1a. This test platform is used by TNO extensively for state estimation and autonomous driving applications. Aside from the normal sensors present on the CAN-bus of the vehicle, TNO has equipped the vehicle with an array of additional sensors, including multiple IMU's and an optical true speed sensor. The measurements are performed on the ATP testtrack [37] as seen in Figure 6.1b. A detailed description of both test vehicle and test track are given in Appendix D.

(a) 2012 Toyota Prius with Corrsys Optical sensor

(b) ATP papenburg testtrack [37]

Figure 6.1: Test Set-up and testing grounds

Reference Measurement

Figure 6.1a illustrates the Toyota Prius test vehicle with the Corrsys optical sensor mounted on the side of the car. This optical sensor is able to measure the true velocity of the vehicle in both longitudinal and lateral direction. As it is mounted to the side of the car, the measured velocities are not identical to the vehicle states V_x and V_y. The rotational velocities also introduce additional components as given by

$$V_{x_{\text{meas}}} = V_x + d_{y_{\text{sens}}} r \tag{6.1}$$

$$V_{y_{\text{meas}}} = V_y - d_{x_{\text{sens}}} r + h_{\text{sens}} \dot{\phi} \tag{6.2}$$

Here $d_{x,y_{\text{sens}}}$ represent the longitudinal and lateral position of the optical sensor with respect to the vehicle's center of gravity, h_{sens} the height at which the sensor is placed and $r, \dot{\phi}$ respectively the yaw and roll rate. After the described correction, the sensor outputs are rotated to compensate for misalignment. This is done with straight line calibration test. It should be noted that in wet road conditions, the optical sensor can sometimes provide inaccurate measurements due to the different reflection of the wet road.

6.2 Filter Tuning

Prior to comparing the behaviour of the different filters under various conditions, a single set of settings has to be found that behaves acceptably under all conditions. Naturally, the settings (noise covariance matrices thresholds and logic thresholds) cannot be tuned for each manoeuvre separately, but should instead work under all circumstances. To this extent, a single set of settings is found that works in both dry and wet conditions and also works in both limit handling and normal driving.

First, thresholds for the innovation logic described in Section 4.3 are tuned. When considering the level of noise present in the measurements the thresholds for $|a_x|$ and $|a_y|$ are selected to be $0.5 m/s^2$. This means that if the magnitude of the measured acceleration does not exceed these thresholds, the innovation is set to zero.

Next, the values related to the error covariance matrix and noise intensities are selected. These values are the initial error covariance matrix $\mathbf{P}_0$ and the measurement and process noise covariance matrices $\mathbf{V}$ and $\mathbf{G}$. When the estimator is initialised, the vehicle is often at rest. In such a case, all vehicle states, with the exception of the augmented friction state, have a relatively small uncertainty. Therefore $\mathbf{P}_0$ is chosen a diagonal matrix, with small initial elements, corresponding with this smaller uncertainty. This also prevents the filter from making excessive measurement based corrections, while both the error and the signal to noise ratio of outputs are still small. The initial error covariance matrix and corresponding standard deviation of each initial state are given by

$$\mathbf{P}_0 = \begin{bmatrix} 1e^{-6} & 0 & \dots & 0 \\ 0 & 1e^{-4} & \ddots & \vdots \\ \vdots & \ddots & 1e^{-4} & 0 \\ 0 & \dots & 0 & 1e^{-3} \end{bmatrix} \Longrightarrow \begin{bmatrix} \sigma_r \\ \sigma_{V_y} \\ \sigma_{V_x} \\ \sigma_{\lambda_\mu} \end{bmatrix} = \begin{bmatrix} 1.0e^{-3} [rad/s] \\ 1.0e^{-2} \quad [m/s] \\ 1.0e^{-2} \quad [m/s] \\ 3.2e^{-2} \quad [-] \end{bmatrix} \tag{6.3}$$

The matrix $\mathbf{V}$ represents the noise intensity of the measurement. However, it should not only describe the noise intensity, but also represent the accuracy of the modelled outputs. To this extent, the estimator model (without output corrections) is compared to the measurements for a manoeuvre that shows good correlation to the model. The magnitude of the resulting innovations is then used to find the matrix $\mathbf{V}$. The process noise matrix $\mathbf{G}$ is then tuned for five different situations to achieve consistent behaviour across all cases. Criteria for the process noise matrix $\mathbf{G}$ is to simultaneously match the error bounds described in Section 4.4 to the actual errors and to have the filter behave in a robust manner.

$$\mathbf{V} = \begin{bmatrix} 5e^{-2} & 0 & \dots & 0 \\ 0 & 1e^{-2} & \ddots & \vdots \\ \vdots & \ddots & 5e^{-5} & 0 \\ 0 & \dots & 0 & 0.5 \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} 2e^{-3} & 0 & \dots & 0 \\ 0 & 2e^{-2} & \ddots & \vdots \\ \vdots & \ddots & 2e^{-3} & 0 \\ 0 & \dots & 0 & 3e^{-6} \end{bmatrix} \tag{6.4}$$

6.3 Estimator results

In contrary to controllers, estimators do not need to be tested in the loop, but can also be tested off-line on logged data. A large set of data is gathered by TNO previously which can be used to this extent. This section presents the results for various driving conditions. Since the model is shown in Section 3.3 to be very accurate for normal conditions, emphasis is placed on more challenging situation such as wet driving, J-turns and hard braking.

6.3.1 Dry Road with J-turn Manoeuvre

The estimators are compared on the dry J-turn manoeuvre. This J-turn manoeuvre is interesting due to the excitation of the vehicle in an oversteering condition. Figure 6.2 illustrates the side slip estimate, lateral acceleration and friction estimate, together with the estimation errors. It can be seen that the model without corrections from the measurements provides a very poor estimate of the body slip angle β. Clearly both EKF and SDRE algorithms are capable of improving the state estimates, resulting in far better β estimates.

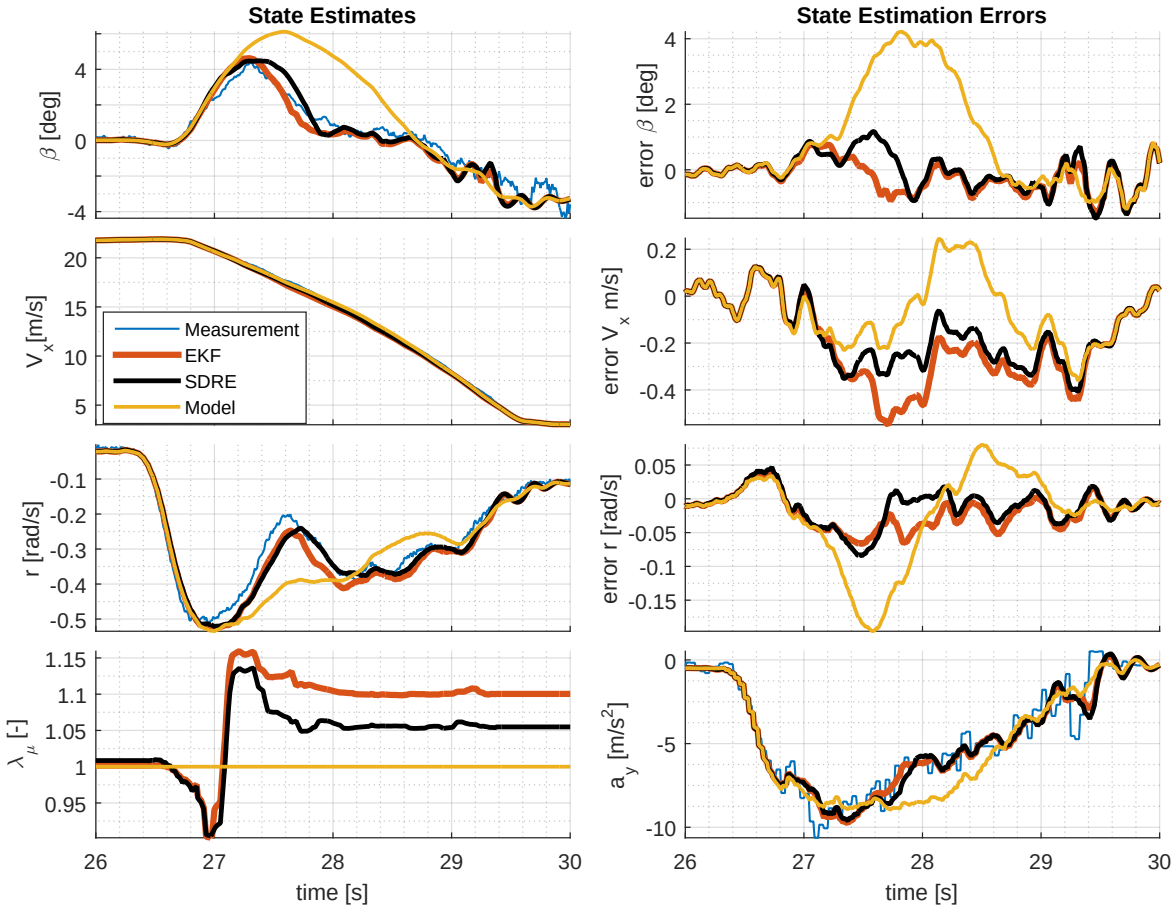


Figure 6.2: Estimator results for right hand J-turn manoeuvre with initial velocity $V_x = 21m/s$

6.3.2 Braking During Friction Change

The estimators are also tested in straight line braking conditions. This experiment is conducted on the braking tracks of the ATP testtrack, which features a transition from asphalt (high friction) to fusion cast basalt (low friction). A test is performed at 60km/h, at which the brake is applied. The results can be seen in Figure 6.3. Note that the optical sensor provides false values due to the reflective properties. This can be identified very easily on the $t \in [15.5s, 16s]$ interval. The forward velocity drops to zero, after which the speed quickly increases again. However, this supposed deceleration and acceleration are not observed in the acceleration measurement. This misinformation by the reference sensor leads to large errors in both body slip angle and forward velocity. This error is thus not very meaningful and should therefore be disregarded.

More interesting in this case is the longitudinal acceleration. It can be seen that once the low friction surface is encountered the longitudinal acceleration a_x prediction has significant errors. This is due to the front wheels transitioning to high slips ($\kappa > 0.5$), for which the model no longer accurately represents the tyre forces under these wet conditions. The calculated brake forces are too high, resulting in a lower estimated velocity and a higher predicted braking acceleration. This discrepancy is detected by the filter and the friction estimate is updated accordingly.

Since the lower limit for V_x is applied before state corrections can occur, the estimator no longer uses the measurements to correct the states for $t > 18s$. Unfortunately no measurements are conducted with a larger approaching speed. It would be interesting to see how the friction estimate would continue for longitudinal excitation over an extended period of time.

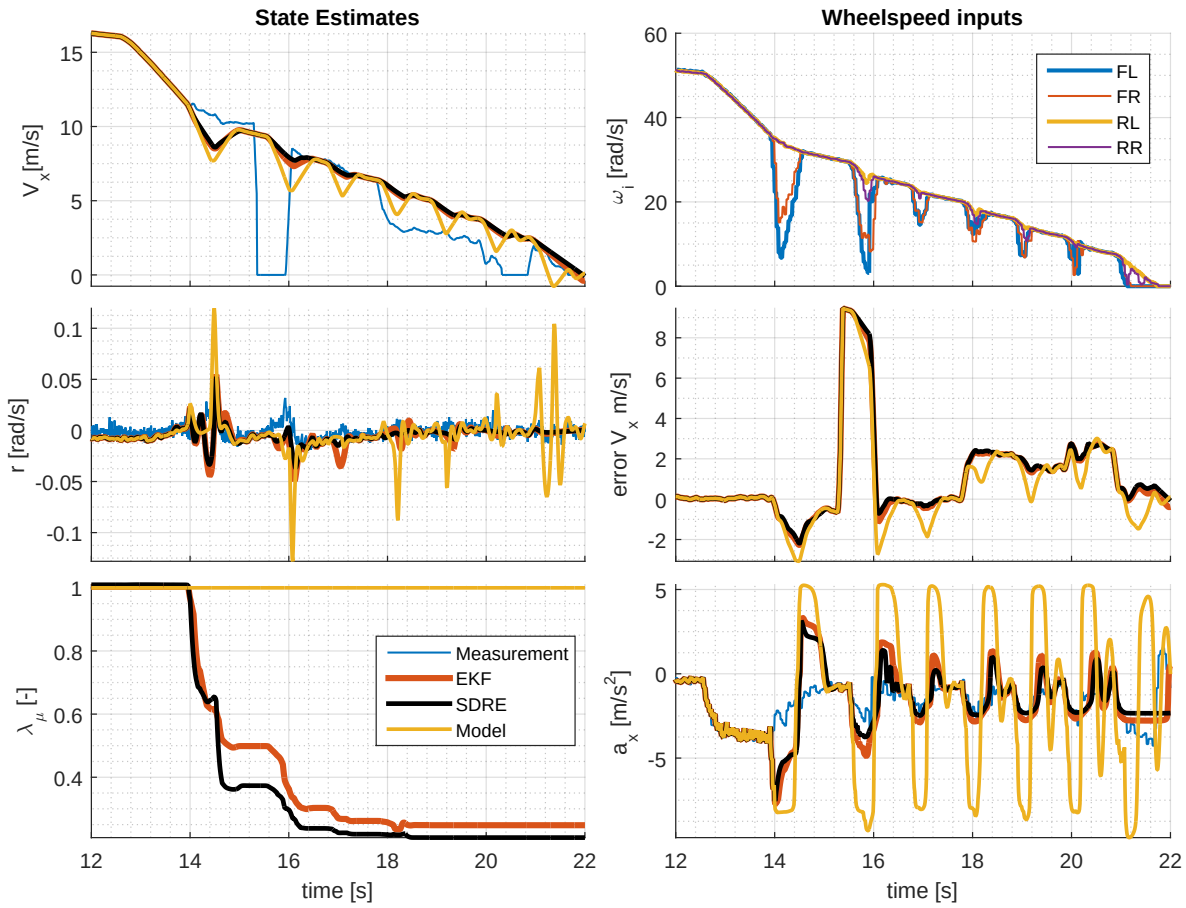


Figure 6.3: Straight line braking for high to low friction transition (Asphalt to Fusion cast basalt). Lane 3 of the Braking Tracks (BMK) [37], transition at $t \approx 13.8s$

6.3.3 Wet Road Handling

To continuously excite the system, measurements are also performed on the wet handling track. Two consecutive laps are driven, to evaluate the values to which the friction estimate converges. Figure 6.4 displays a zoom of the results of this longer run. Again, the false values from the optical sensor can be observed in both the forward velocity and side slip estimates. It can be seen that the filter converges again to the lower friction value, as is the case for the braking experiment. However, it should be noted that it is capable of doing so without the large differences in output that were present in the braking experiment. The results of all state estimates for the entire experiment are given in Appendix E.

Additionally, the same dataset is also used for a case where the estimator model is non-convex (shape factor $C_y = 1.3$)(see Figure B.2 to see the effect of parameters on characteristics). This non-convex model is shown in Chapter 2 to potentially lead to problems, due to multiple distinct states resulting in similar outputs. The results for this non-convex estimator model can also be found in Appendix E.2. The comparison in the appendix demonstrates that both the EKF and SDRE estimator are still capable of accurately tracking the states.

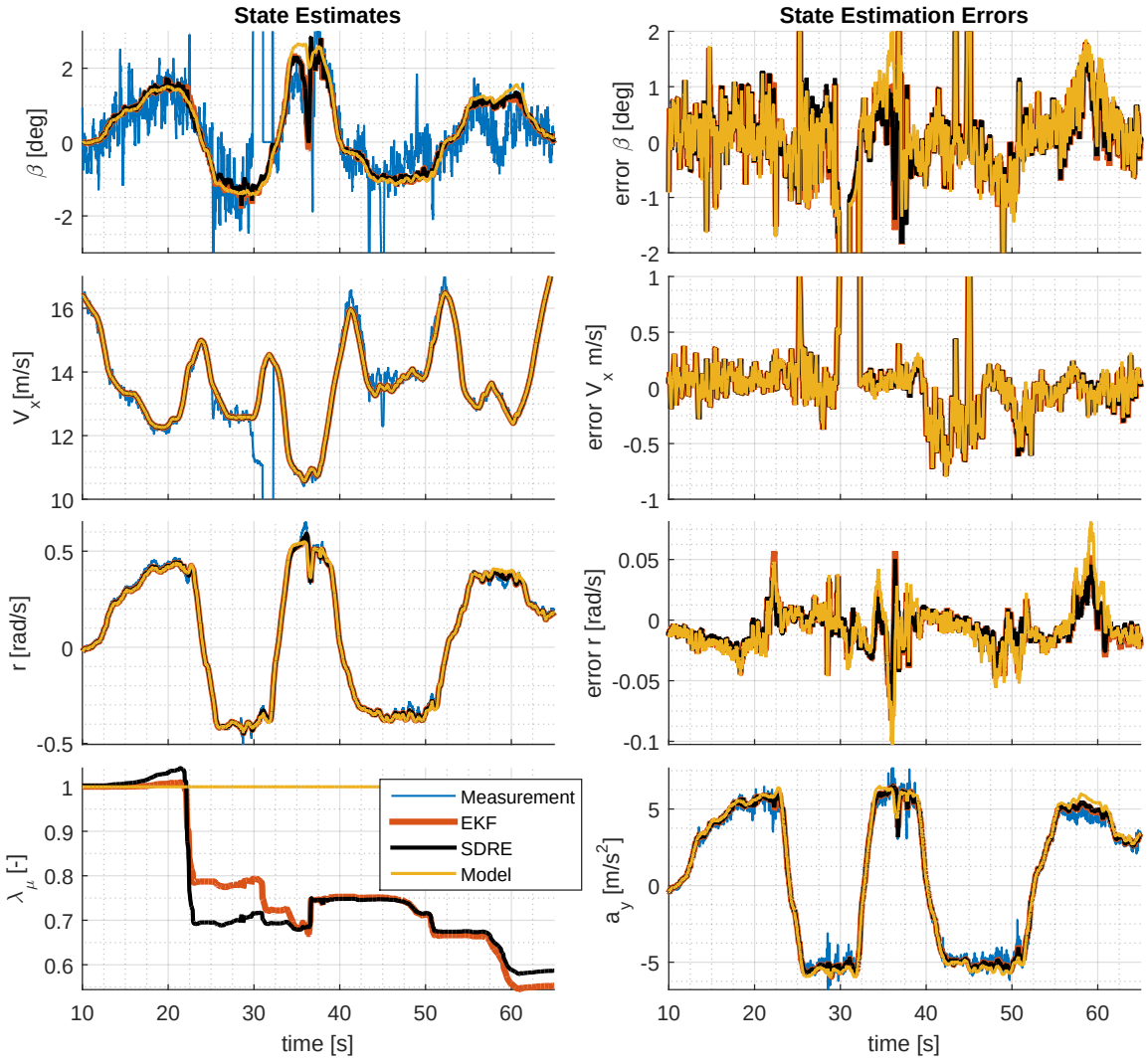


Figure 6.4: Wet handling track (NHK) [37] driving. Forward velocity V_x between 10 and 16 m/s

6.4 Comparison of the Estimators with TNO Estimator

Some datasets are available on which the TNO VSE is tested. Due to confidentiality, only the state estimates of the VSE are available, without the predicted outputs. The datasets can be used to compare the performance of the developed estimators with the TNO VSE. A J-turn, a severe lane change and a sinusoidal input are the three case that will be used to compare the estimators.

6.4.1 Lane Change

A lane change with limited lateral acceleration is performed. It can be seen in Figure 6.5 that the developed estimators hardly need correction, as the vehicle model already provides accurate estimates of the body slip angle. The TNO VSE seems to suffer from the selection of incorrect vehicle parameters, since the body slip angles are not accurately described for these relatively low lateral accelerations.

6.4.2 Constant Radius Cornering

A constant radius corner with increasing velocity is used for comparison. The results can be found in Figure 6.6. Again it can be seen that the TNO VSE does not describe the side slip angle accurately. Additionally, the TNO VSE seems to have some additional logic for low speeds. It can be seen that the test is initiated at standstill. Although not illustrated, the steer angle is initially non-zero. Therefore it is expected to see a constant slip angle as soon as the car starts moving. The developed estimators do show this results, whereas the TNO VSE gradually converges to this slip angle in the course of two seconds.

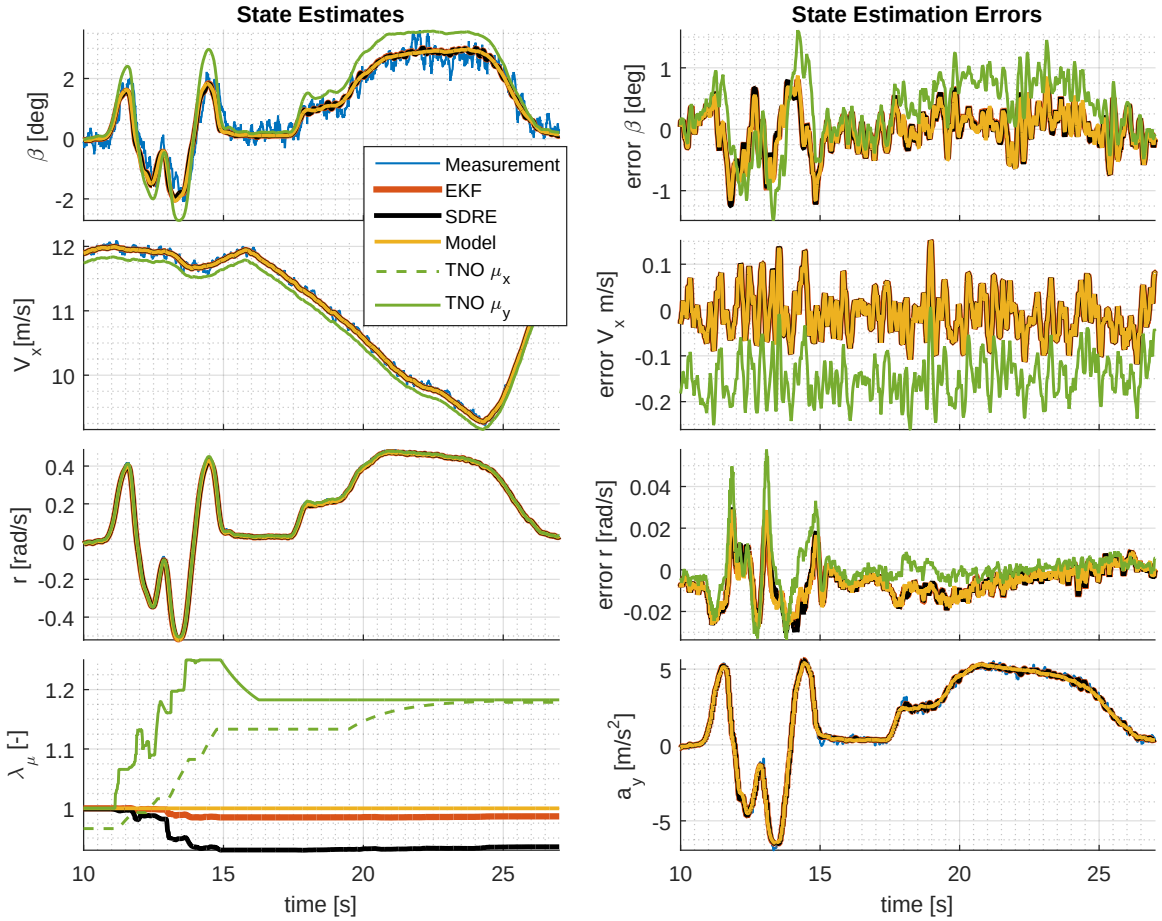


Figure 6.5: Lane Change manoeuvre at forward velocity of $V_x \approx 12m/s$

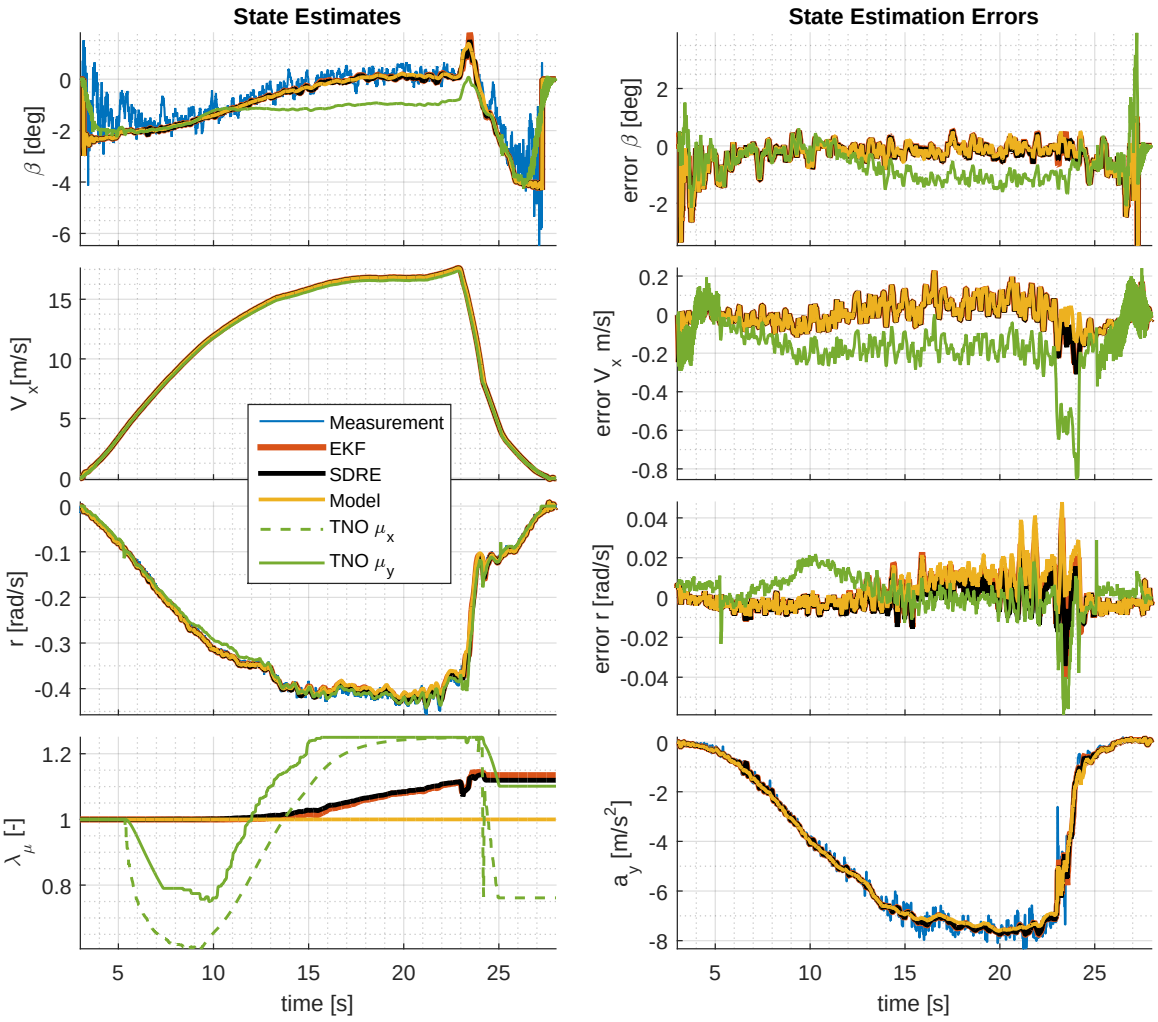


Figure 6.6: Constant radius corner $R \approx 42.5$ for increasing velocity. Maximum forward velocity of $V_x \approx 17m/s$

6.4.3 J-turn

Finally, the J-turn manoeuvre is considered. Figure 6.7 and Figure 6.8 show the results for respectively a left and right hand J-turn at different forward velocities. Again the developed estimators provide more accurate state estimates than the TNO estimator, as seen in the estimation error plots. Interestingly, the TNO VSE seems to be more accurate in estimating the yaw rate, perhaps due to a lower measurement noise intensity selected than the developed estimators.

It is also worth noting that in all cases presented involving J-turn manoeuvres, the model without measurement based corrections does not provide accurate results for the slip angle. This can significantly be improved by tuning the roll stiffness distribution parameter, which in combination with the influence of vertical load on cornering stiffness changes the balance of the vehicle. It is interesting to see the conflicting influence in both cases. Perhaps the roll damping distribution is not identical to the roll stiffness distribution, such that the behaviour in transients is significantly different than in steady-state cornering. Potentially, this could be modelled better when using a roll axis model. Due to time limitations this is however not further investigated and is left for future research.

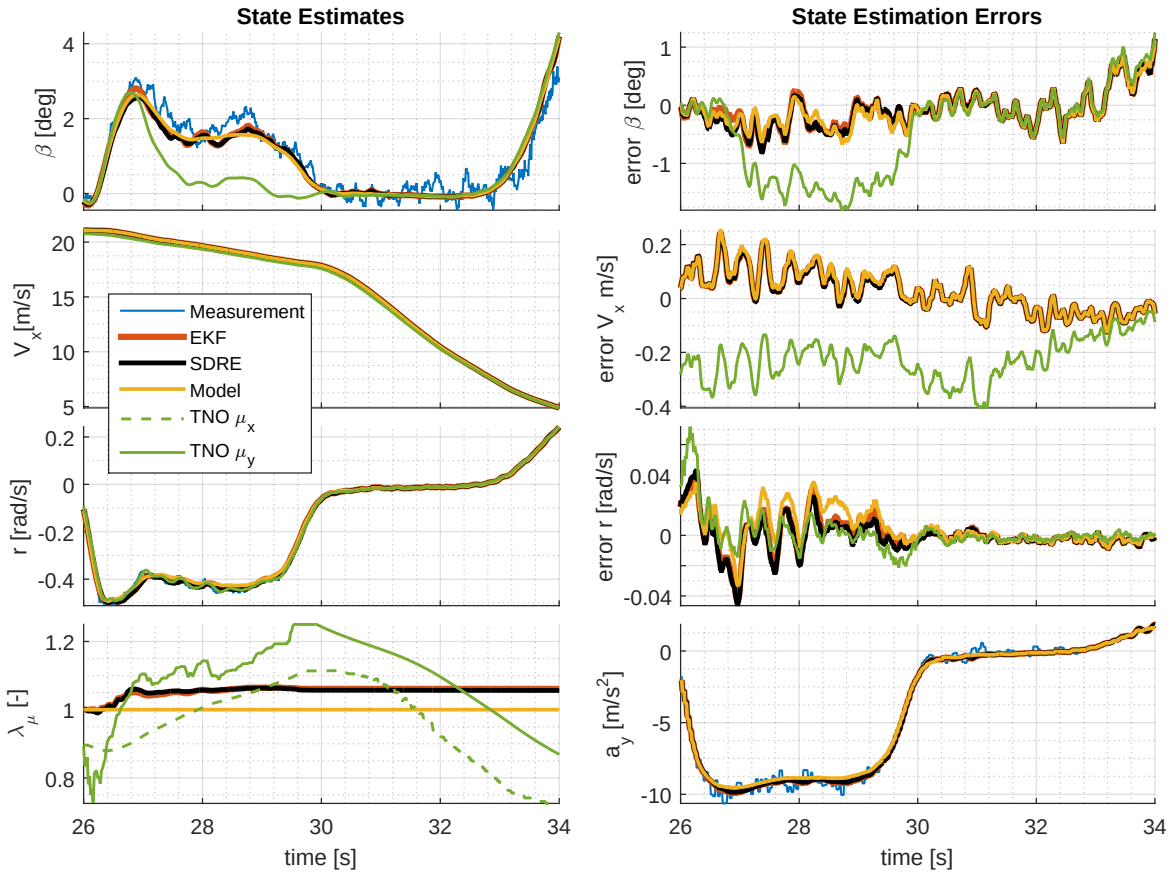


Figure 6.7: Right hand J-turn manoeuvre at forward velocity of $V_x \approx 21m/s$

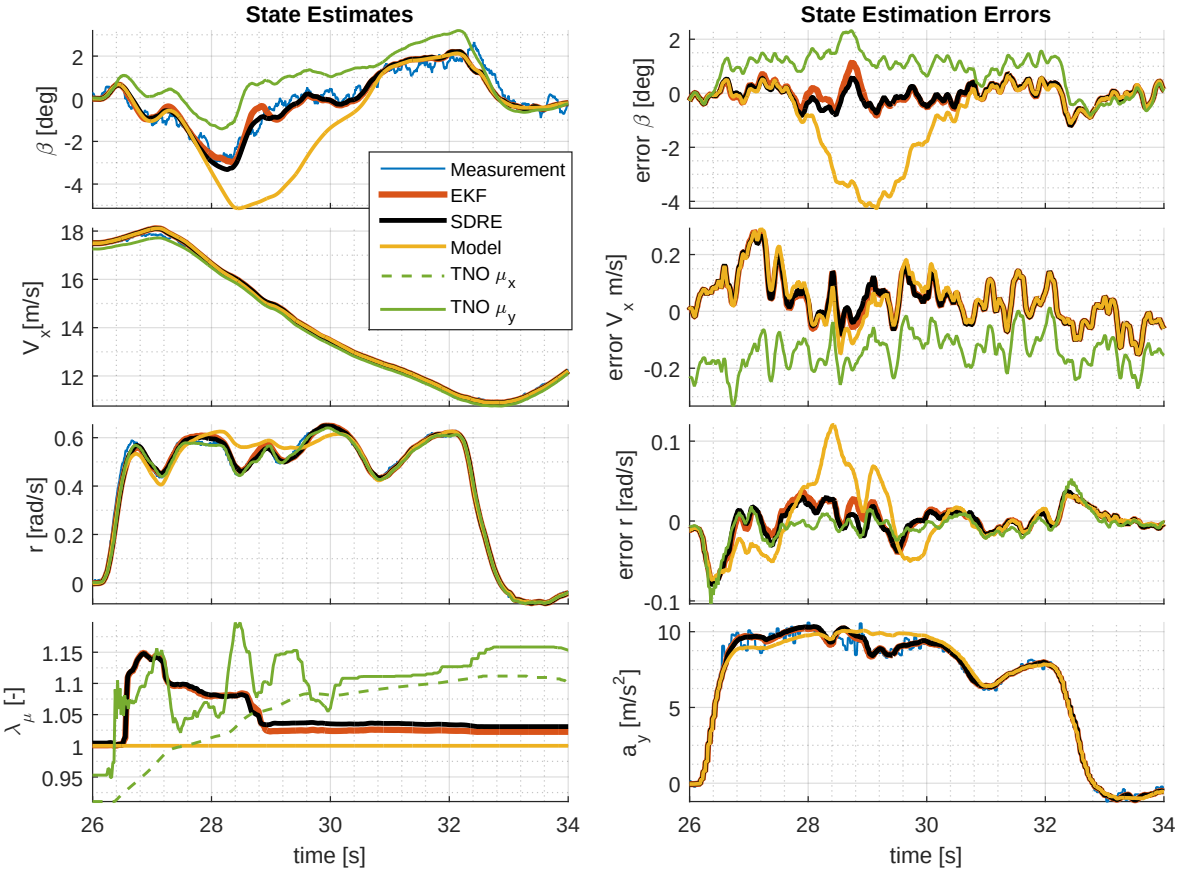


Figure 6.8: Left hand J-turn manoeuvre, at forward velocity of $V_x \approx 18m/s$

6.4.4 Sinusoidal Steering Input

A sinusoidal input of approximately 0.4 Hz is applied at approximately 65km/h. The resulting states and state estimates are given in Figure 6.9. Interestingly, all estimators seem to have difficulties describing the body slip accurately. Both estimators developed in this project seem to have some 'overshoot' for negative slip angles. Moreover, the TNO VSE even seems to suffer from a phase delay. Possibly, this is due to the roll dynamics becoming relevant at these input frequencies. Also the bank angle that is estimated in parallel for the TNO VSE could play a role here. However, since this bank angle estimate is unavailable this could not be checked.

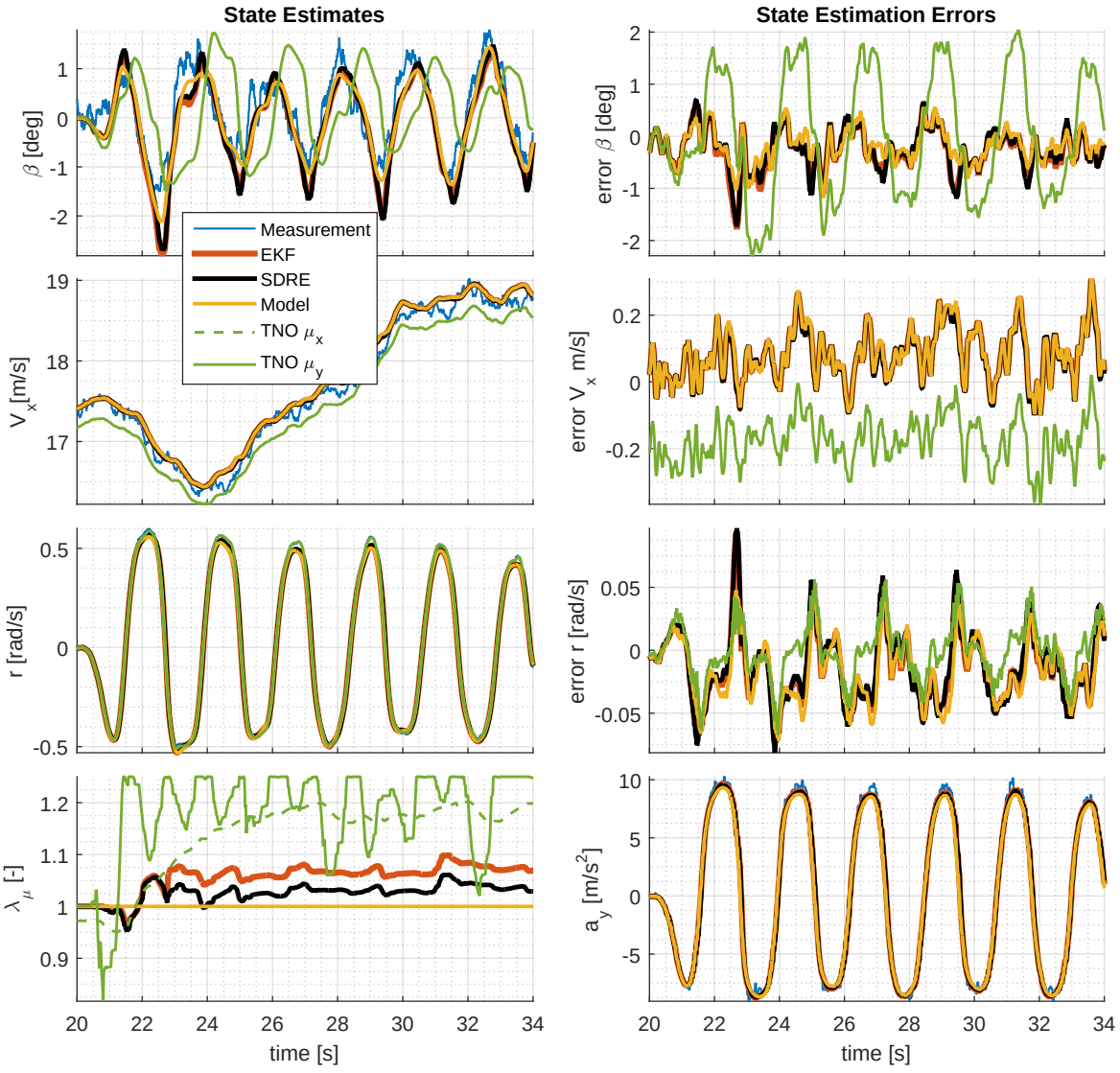


Figure 6.9: Sinusoidal steering input with maximum lateral acceleration at forward velocity of $V_x \approx 17m/s$

6.5 Real-time Implementation

Aside from testing the developed estimator on previously logged data, the VSE is also implemented on the real-time platform, speedgoat [38], present in the Toyota Prius used by TNO. The main objective is to validate that the developed estimator is capable of running in real-time in terms of computational load. Additionally, it is of interest how the estimator

behaves under normal city driving, when excitation is continuously low for an extended period of time. Additionally, it is interesting to see how the estimator behaves for special conditions like start-stop situations and reversing. Naturally, since wheel speed signals are non-directional, reverse driving is not tracked properly. However, when returning to normal forward driving, the estimator is able to again track the system. Although not implemented during this project, the issue with reverse driving could be solved by also using the selected gear as an input. Normal city driving showed consistent behaviour throughout, with similar behaviour to that of the TNO VSE.

6.6 Summary and Discussion

The developed EKF and SDRE estimator are both tested under various conditions. The estimators are capable of detecting changes in friction in both lateral and longitudinal situations. The SDRE and EKF estimators show very similar estimates for all presented cases.

Note that the presented estimators both only estimate the ultimate friction. However, as presented in [39], changing road conditions does not only change the ultimate friction, but often also influences the stiffness properties of the tyres. A model as presented in [39] can be used to identify changing conditions under lower excitations. It is interesting to investigate these possibilities. However, due to the scope of this project emphasis is only placed on the comparison of the SDRE and EKF algorithms.

Conclusions regarding which estimator principle is better, must be carefully drawn. Both estimators seem to perform very similarly. However, the effort in implementation is not taken into account when only looking at the results. As also mentioned in Chapter 2, the SDC form used by the SDRE estimator is non-unique. However, finding an SDC form that results in good performance is not trivial. In fact, the selection of the SDC form proved to be critical for good performance of the estimator. The final SDC form that strongly resembles the shape of the Jacobian presented by the EKF proves to be the best performing form. Additionally, finding correct values for noise intensities that result in reliable estimates proved to be easier. The main benefit of the SDRE estimator seems to be the computational effort, the SDRE estimator only needing approximately 21.7 % of the computation time required by the EKF estimator. This can have significant benefits when implemented on commercial systems with lower computational power.

Additionally, the developed estimators are shown to provide better estimates of side slip angle β. Arguably, the friction estimates are also better. This statement can not be truly validated, since the true friction is not available as reference. However, it is far more likely that the friction coefficient is approximately 1 on the dry surface that the experiments were performed on. This is seen in the results by the estimators developed during this project. In contrast, the TNO VSE fluctuates between 0.75 and 1.25 (even friction estimates of 0.6 are seen for lower speeds), which seems a very unlikely variation for normal dry asphalt.

Although these results are promising, it can not yet be concluded that the newly developed estimators are better than the TNO VSE. The inaccuracy of the TNO VSE seems far more likely to be contributed to parameter tuning than fundamental model limitations. Additionally, it should be noted that the TNO VSE is also capable of estimating road bank angle in parallel. This additional degree of freedom greatly influences the complexity of the estimator. Nevertheless, for the datasets that are available, the TNO VSE is outperformed by the developed estimators in every case.

Chapter 7

Conclusion and Recommendations

7.1 Conclusion

During this thesis a new vehicle state estimator is developed capable of simultaneously estimating the vehicle's body slip angle and vehicle-road friction. This estimator is based on a model prediction in combination with measurement correction. The information from prediction and measurement are combined using either the commonly used EKF or novel SDRE algorithm. An analytic comparison of both EKF and SDRE estimators is presented using simple examples. Potential advantages of the SDRE algorithm are illustrated using examples taken from literature. Additionally, the various steps in the algorithms are isolated and analysed separately, to study the convergence properties for a static scalar system. Clear differences in convergence properties between the EKF and SDRE are identified.

A vehicle model is created and a parameter identification is performed to describe the dynamics of a 2012 Toyota Prius. The prediction model is shown to be accurate in most conditions. Derivation of the Jacobian matrices, used in the EKF estimator, is presented for this vehicle model. Similarly design of the State Dependent Coefficients, used in the SDRE algorithm is presented. It is demonstrated that these SDC forms are non-unique, and as such, there is more design freedom in finding them. Multiple SDC forms are derived and tested. The best performing SDC form is found using a heuristic approach in which properties of the Jacobian matrix used in the EKF are mimicked. For fair comparison, both the EKF and SDRE algorithms are implemented and compared using an identical vehicle model and identical addition supportive logic. Additionally, a single set of settings for logic thresholds and noise intensities is selected for both estimators. Simulations, as well as experiments, are used to compare the estimators in various conditions. Additionally, the estimators are implemented in the real-time platform in the Toyota Prius.

Simulations are used to evaluate the behaviour of the estimators for friction estimation since no reference for the true friction is available in experiments. Steady-state cornering in which friction is either dropped or increased is used to evaluate the convergence speed of the estimators. The EKF and SDRE based estimators both display very similar convergence properties. Logic is also implemented to aid in detecting sudden changes in friction. When such a change is detected, the correction terms for friction are increased. This results in significantly faster convergence of the friction estimate and consequently smaller errors in the remaining state estimates. Additionally, the robustness of the estimators is tested for parameter uncertainty. Two cases are studied in which respectively the vehicle mass and tyre shape factor are chosen differently. The different mass represents cases where additional passengers and load are present in the vehicle. The estimators are shown to be capable of showing accurate state estimates under these conditions.

To analyse the estimators with real-world data, datasets generated at the Automotive Testing Papenburg test track are used. Exceptions are highly dynamics manoeuvres, such as a J-turn,

where transient effects become more dominant. These cases, where the stand-alone model did not provide accurate predictions, are successfully corrected via measurements using the EKF and SDRE algorithms. Both EKF and SDRE are capable of estimating the states with similar accuracy, when filter settings (eg. noise intensities and initial conditions) are correctly tuned. The selection of these noise intensities proved critical in the filters convergence speed and robustness. Both dry and wet road conditions are tested in the experiments. Although no reference for the friction is available for the experiments, the tests performed in wet conditions did show that the friction estimates changed. In contrast, in dry conditions the friction estimate remained approximately constant as is to be expected.

Estimates of the TNO VSE are only available for some limited datasets. These datasets are used to compare the developed estimators with the TNO VSE. The newly developed estimators showed more accurate estimates for both friction and side slip angle, for all tested manoeuvres and conditions. These results however must be critically interpreted, as the TNO VSE is most likely sub-optimally tuned for this specific vehicle. Additionally, the TNO VSE is also capable of simultaneously estimating the road bank angle, whereas the newly developed estimators are not. The addition of this road bank angle estimate greatly increases the difficulty in accurately estimating all states. Nevertheless, for the conditions in which the tests are performed, the developed estimators are shown to be more accurate in all cases.

In conclusion, the EKF and SDRE result in very similar behaviour. Both in simulations as in experimentations, the resulting state estimates are in general very similar. However, the added benefit of the SDRE proved to be the computational load, which is only 21% of the computational load required by the EKF. This could prove beneficial in commercial applications where computational power of the hardware is limited.

7.2 Recommendations

To further improve the developed estimators the following adaptations are proposed. A brief explanation will be given to motivate each suggestion.

Estimator Model Including the wheel dynamics might reduce the noise present on the wheelspeed input signals. These additional dynamics can also be utilized to detect friction changes faster for the longitudinal case. This could help to improve the overall accuracy of the estimator.

In the current estimator, steady state roll corrections are applied to compensate for larger lateral acceleration measurements. Although this works, a cleaner solution would be to model roll and pitch motions. This could be used to improve the accuracy of the outputs and improve the accuracy of the model in transient conditions. Additionally, this eliminates the algebraic loop in calculations of the vertical loads.

Including the pitch and roll dynamics can also be used in combination with roll and pitch rate sensors, which are currently not utilized. This can be used to include the road slope and bank angle estimation. With road slope and bank angle estimation included, the conditions in which the estimator provides accurate results likely increases.

Algorithm During the development of the SDRE based estimator, it is found that the selected SDC form is critical to the performance of the estimator. Although a suitable form is found and used, further investigation of different SDC forms can prove to be useful.

Both EKF and SDRE estimators are capable of tracking the desired state estimates, even in the cases of parameter uncertainty. However, no rigorous stability proof is presented for either case. A more rigorous approach in stability analysis could also prove useful in determining the best performing, or most robust, SDC form.

Appendix A

Probability and Observability

This appendix will very briefly summarise the background and algorithm steps of Kalman filtering.

A.1 Expected Value and Conditional Probability

This section will summarize the background of Bayesian filtering. Probability and conditional probability will be explained. Also properties of the expected value operator will be provided. For more detailed and thorough explanation the reader can refer to [17, 25].

A.1.1 Conditional Probability

Probability theory is at the heart of many filtering algorithms. Therefore the bases will be briefly explained. Consider a random variable x. The probability that this random variable attains a certain value is given by the probability density function $p(x)$. Properties of this function are the following

$$p(x) \geq 0, \qquad \int_{-\infty}^{\infty} p(x)dx = 1 \tag{A.1}$$

Since probabilities are always larger or equal to zero, and the total probability of all possible values is always one. Consider now a second random variable z. The probability of simultaneously finding both x and z is denoted with $p(x, z)$. The information of simultaneously finding both x and z can be used if one of them is known.

$$p(x|z) \triangleq \frac{p(x, z)}{p(z)} \tag{A.2}$$

Where $p(x|z)$ denotes the conditional probability density function. This indicates the probability of random variable x given random variable z. Conditional probability is at the base of many filtering applications. Important here is Bayes theorem as found in [17] which will be briefly explained here. $p(x, z)$ can be expanded in two ways.

$$p(x, z) = p(x \mid z)p(z) \tag{A.3}$$
$$p(x, z) = p(z \mid x)p(x) \tag{A.4}$$

(A.3) gives the probability of variable x for a given z which is multiplied with the probability of that same z. Similarly, (A.4) gives the probability of variable z for a given x which is multiplied

with the probability of that same x. Both result in the probability of encountering both x and z. When rearanging the terms we find the result of Bayes Theorem

$$p(x \mid z) = \frac{p(z \mid x)p(x)}{p(z)} \tag{A.5}$$

Interpretation of (A.5) will now be explained. Consider state x and observations z, also consider there is some *a priori* knowledge about the probability distribution of state x ($p(x)$). It can now be modelled what the probability is of making observations z for a certain x ($p(z|x)$). If an observation is now obtained, it can be used to obtain more knowledge about the precise state x via (A.5). This results in the probability of state x given an observation z ($p(x|z)$). Finally, the result is normalised by the total probability of observation z ($p(z)$). Thus (A.5) provides a powerful method of combining prior knowledge with new observations to obtain a more accurate state estimate.

Expected value operator

In estimation, the expected value is often used. Following [17] the following operator is defined

$$E[x] \triangleq \int_{-\infty}^{\infty} xp(x)dx \tag{A.6}$$

Conditional expectation is now defined as

$$E[x \mid z] \triangleq \int_{-\infty}^{\infty} xp(x \mid z)dx \tag{A.7}$$

Of specific interest are now $E[x^n]$ and $E[(x-\overline{x})^n]$ also referred to as the n^{th} moment and the n^{th} central moment respectively. In particular the first moment and second central moment provide useful values

$$\overline{x} = E[x] \tag{A.8}$$
$$\sigma_x = E[(x-\overline{x})^2] \tag{A.9}$$
$$\tag{A.10}$$

where $\overline{x}$ and σ_x denote respectively the mean and variance of random variable x.

For non-scalar values the following notation is used

$$\overline{\mathbf{x}} = E[\mathbf{x}] \tag{A.11}$$
$$\mathbf{P_{xx}} = E[(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}}] \tag{A.12}$$

With $\mathbf{P_{xx}}$ the variance of $\mathbf{x}$.

A.2 Non-linear Observability

Consider the system given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}(t) + \mathbf{\Gamma}(\mathbf{x})\mathbf{w}(t) \tag{A.13}$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}(t) \tag{A.14}$$

An observer can only reconstruct the states based on measurements if the system is said to be observable. A system is observable if and only if it is possible to reconstruct any initial state $\mathbf{x}(0)$ using only a finite amount of measurements. For general non-linear systems as defined in (A.13) and (A.14) the system is observable in the general non-linear sense if the following is true according to [7]:

$$rank \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}}\mathbf{h}(\mathbf{x}) \\ \frac{\partial}{\partial \mathbf{x}}L_f(\mathbf{h}(\mathbf{x})) \\ \vdots \\ \frac{\partial}{\partial \mathbf{x}}L_f^{n-1}(\mathbf{h}(\mathbf{x})) \end{bmatrix} = n \tag{A.15}$$

In which $L_f(\mathbf{h}(\mathbf{x}))$ represents the Lie derivative of $\mathbf{h}(\mathbf{x})$ along vector field $\mathbf{f}(\mathbf{x})$ defined as

$$L_f^n(\mathbf{h}(\mathbf{x})) = L_f(L_f^{n-1}\mathbf{h}(\mathbf{x})) \tag{A.16}$$

$$L_f(\mathbf{h}(\mathbf{x})) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) \tag{A.17}$$

For linear systems this reduces to the well know observability matrix (A.20), since

$$\mathbf{f}(\mathbf{x}) = \mathbf{F}\mathbf{x}, \qquad \mathbf{h}(\mathbf{x}) = \mathbf{H}\mathbf{x}, \qquad \frac{\partial}{\partial \mathbf{x}}\mathbf{h}(\mathbf{x}) = \mathbf{H}$$

then

$$L_f(\mathbf{h}(\mathbf{x})) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) = \mathbf{H}\mathbf{F}\mathbf{x} \tag{A.18}$$

$$L_f^2(\mathbf{h}(\mathbf{x})) = \left(\frac{\partial L_f(\mathbf{h}(\mathbf{x}))}{\partial \mathbf{x}}\right)\mathbf{f}(\mathbf{x}) = (\mathbf{H}\mathbf{F})\mathbf{F}\mathbf{x} = \mathbf{H}\mathbf{F}^2\mathbf{x} \tag{A.19}$$

And so on. Resulting in elements of (A.15) of

$$\frac{\partial}{\partial \mathbf{x}}L_f(\mathbf{h}(\mathbf{x})) = \mathbf{H}\mathbf{F}, \qquad \frac{\partial}{\partial \mathbf{x}}L_f^2(\mathbf{h}(\mathbf{x})) = \mathbf{H}\mathbf{F}^2, \qquad \frac{\partial}{\partial \mathbf{x}}L_f^{n-1}(\mathbf{h}(\mathbf{x})) = \mathbf{H}\mathbf{F}^{(n-1)}$$

which then in turn results the observability matrix and corresponding observability criteria, which is known from linear system theory.

$$rank(\mathcal{O}) = rank \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\mathbf{F} \\ \vdots \\ \mathbf{H}\mathbf{F}^{n-1} \end{bmatrix} = n \tag{A.20}$$

A.3 Covariance Propagation

As outlined in Section 2.3, the propagation of uncertainty through the system assumed by EKF and SDRE is only an approximation. This section will provide the figures to support this claim.

To investigate the propagation of uncertainty an initial Gaussian distribution is monitored as it propagates through several time-steps. The Gaussian distribution is represented by a set of $N = 1e5$ samples, drawn from the distribution. These samples provide the initial conditions of $N = 1e5$ short simulations.

It can be seen from the top left bar plot in Figure A.1 that after even three time-steps, the distribution is no longer Gaussian when propagated through this non-linear system. Similar observations are made in [26], which presents this as a problem with the Extended Kalman filter. The proposed unscented Kalman filter should give a more accurate description of the error covariance.

The Extended and Unscented Kalman filter together with the SDRE based algorithm are compared in describing the distribution. Here, only the mean of the initial distribution is propagated through the system, together with error covariance matrix $\mathbf{P}$. In propagation of the error covariance matrix only initial uncertainty is propagated through (A.21). As can be recognised from the equation no process noise is included and no observations on the system are made to reduce the uncertainty. The resulting error covariance is then used each time step to generate a new Gaussian distribution, to represent the distribution in the plot.

$$\dot{\mathbf{P}} = \mathbf{AP} + \mathbf{PA}^{\mathbf{T}} \tag{A.21}$$

A.4 Discrete Kalman Filter Equations

Extended Kalman filter equations are also presented her for completeness in discrete time as given by [23].

$$\mathbf{A}_{[\mathbf{i},\mathbf{j}]} = \left.\frac{\partial \mathbf{f}_{[i]}}{\partial \mathbf{x}_{[j]}}\right|_{(\hat{\mathbf{x}})}, \quad \mathbf{H}_{[\mathbf{i},\mathbf{j}]} = \left.\frac{\partial \mathbf{h}_{[i]}}{\partial \mathbf{x}_{[j]}}\right|_{(\hat{\mathbf{x}})} \tag{A.22}$$

Time update

$$\hat{\mathbf{x}}_{\mathbf{k}}^{-} = \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{k-1}}^{-}, \mathbf{u}_{\mathbf{k-1}}) \tag{A.23}$$

$$\mathbf{P}_{\mathbf{k}}^{-} = \mathbf{A}(\hat{\mathbf{x}}_{\mathbf{k-1}}^{-})\mathbf{P}_{\mathbf{k-1}}\mathbf{A}(\hat{\mathbf{x}}_{\mathbf{k-1}}^{-})^{\mathbf{T}} + \mathbf{G} \tag{A.24}$$

Measurement update

$$\mathbf{K}_{\mathbf{k}} = \mathbf{P}_{\mathbf{k}}^{-}\mathbf{H}(\hat{\mathbf{x}}_{\mathbf{k}}^{-})^{\mathbf{T}}\left(\mathbf{H}(\hat{\mathbf{x}}_{\mathbf{k}}^{-})\mathbf{P}_{\mathbf{k}}^{-}\mathbf{H}(\hat{\mathbf{x}}_{\mathbf{k}}^{-})^{\mathbf{T}} + \mathbf{R}\right)^{-\mathbf{1}} \tag{A.25}$$

$$\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k}}^{-} + \mathbf{K}_{\mathbf{k}}\left(\mathbf{z}_{\mathbf{k}} + \mathbf{h}(\hat{\mathbf{x}}_{\mathbf{k}}^{-}, \mathbf{0})\right) \tag{A.26}$$

$$\mathbf{P}_{\mathbf{k}} = (\mathbf{I} - \mathbf{K}_{\mathbf{k}}\mathbf{H}(\hat{\mathbf{x}}_{\mathbf{k}}^{-}))\mathbf{P}_{\mathbf{k}}^{-} \tag{A.27}$$

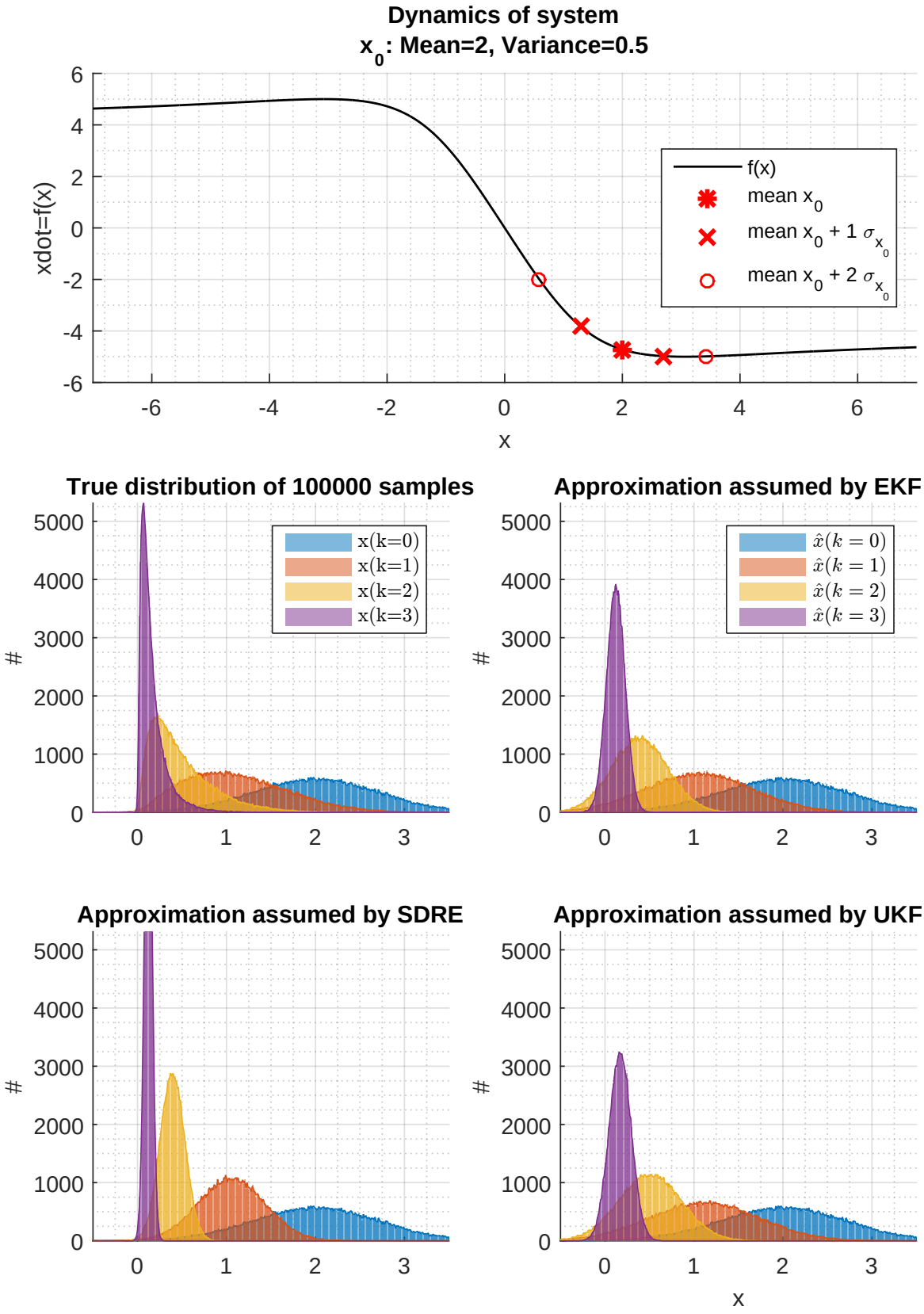


Figure A.1: Comparison of uncertainty propagation through the system for various algorithms

Appendix B

Tyre Model

This appendix lists all equations and parameters involved in calculating the forces developed by the tyres. Equations used are taken from [28].

B.1 Equations

B.1.1 Relaxation Length

Slip speeds are calculated using a first order system with time constant τ.

$$\tau_y \frac{dv_{sy}^*}{dt} + v_{sy}^* = v_{sy}, \qquad \tau_y = \frac{s_x}{v_x} \tag{B.1}$$

$$\tau_x \frac{dv_{sx}^*}{dt} + v_{sx}^* = v_{sx}, \qquad \tau_x = \frac{s_y}{v_x} \tag{B.2}$$

B.1.2 Slip and General Parameters

General slip definitions are provided here. Note that symmetric (braking and driving) longitudinal slip characteristics are used, in absence of detailed tyre measurements.

$$\alpha = \frac{-V_{sy}^*}{V_x}, \qquad \kappa = \frac{-V_{sx}^*}{V_x} \tag{B.3}$$

$$\sigma_x = \frac{\kappa}{1+|\kappa|}, \qquad \sigma_y = \frac{\tan(\alpha)}{1+|\kappa|}, \qquad \sigma = \sqrt{\sigma_x^2 + \sigma_y^2 + \epsilon} \tag{B.4}$$

$$d_{fz} = \frac{F_z - F_{z_{nom}}}{F_{z_{nom}}} \tag{B.5}$$

B.1.3 Pure Slip Coefficients

$$C_y = P_{CY1}, \qquad C_x = P_{CX1} \tag{B.6}$$

$$\mu_y = (P_{DY1} + P_{DY2} d_{fz})\lambda_{\mu_y}, \qquad \mu_x = (P_{DX1} + P_{DX2} d_{fz})\lambda_{\mu_x} \tag{B.7}$$

$$D_y = F_z \mu_y, \qquad D_x = F_z \mu_x \tag{B.8}$$

$$E_y = P_{EY1} + P_{EY2} d_{fz}, \qquad E_x = P_{EX1} + P_{EX2} d_{fz} \tag{B.9}$$

$$K_y = P_{KY1} F_{z_{nom}} \sin(2\text{atan}(\frac{F_z}{P_{KY2} F_{z_{nom}}}))\lambda_{K_y}, \qquad K_x = F_z(P_{KX1} + P_{KX2} d_{fz})\lambda_{K_x} \tag{B.10}$$

$$B_y = \frac{K_y}{C_y D_y}, \qquad B_x = \frac{K_x}{C_x D_x} \tag{B.11}$$

B.1.4 Combined Slip Forces with Similarity Method

$$F_{y0}(\sigma) = D_{mf_y} \sin[C_{mf_y} \arctan\{B_{mf_y}\sigma - E_{mf_y}(B_{mf_y}\sigma - \arctan B_{mf_y}\sigma)\}] \tag{B.12}$$

$$F_{x0}(\sigma) = D_{mf_x} \sin[C_{mf_x} \arctan\{B_{mf_x}\sigma - E_{mf_x}(B_{mf_x}\sigma - \arctan B_{mf_x}\sigma)\}] \tag{B.13}$$

$$F_y = F_{y0}\frac{\sigma_y}{\sigma} \tag{B.14}$$

$$F_x = F_{x0}\frac{\sigma_x}{\sigma} \tag{B.15}$$

B.1.5 Aligning Moment

$$D_t = F_z \frac{R_0}{F_{z_{nom}}}(Q_{DZ1} + Q_{DZ2}d_{fz})\text{sign}(V_x) \tag{B.16}$$

$$B_t = (Q_{BZ1} + Q_{BZ2}d_{fz})\frac{\lambda_{K_y}}{\lambda_{\mu_y}} \tag{B.17}$$

$$C_t = Q_{CZ1} \tag{B.18}$$

$$E_t = (Q_{EZ1} + Q_{EZ2}d_{fz})(1 + Q_{EZ4}\frac{2}{\pi}\text{atan}(B_tC_t\alpha)) \tag{B.19}$$

$$\alpha_{eq} = \text{sign}(\alpha)\sqrt{\alpha^2 + \left(\frac{K_x}{K_y}\right)^2 \kappa^2} \tag{B.20}$$

$$L_t = D_t \cos(C_t\text{atan}(B_t\alpha_{eq} - Et * (Bt * \alpha_{eq} - atan(Bt * \alpha_{eq})))) \tag{B.21}$$

B.1.6 Effective Rolling Radius

$$R_\Omega = R_0\left(Q_{RE0} + Q_{V1}\left(\frac{\omega R_0}{V_0}\right)^2\right) \tag{B.22}$$

$$R_e = R_\Omega - \frac{F_{z_{nom}}}{C_z}\left(D_{R_{eff}} \arctan\left(B_{R_{eff}}\frac{F_z}{F_{z_{nom}}}\right) + F_{R_{eff}}\frac{F_z}{F_{z_{nom}}}\right) \tag{B.23}$$

B.2 Tyre Parameters

Table B.1: Tyre parameters used for 2012 Prius

parameters	**value**	**designation**	**source**
General			
$F_{z_{nom}}$	4000	Nominal load	average between front and rear
R_0	0.317	Unloaded Radius	205 55R16
s_x	0.3	relaxation length x	fitted
s_y	0.3	relaxation length y	fitted
Scaling			
λ_{μ_x}	1	scaling longitudinal friction	
λ_{μ_y}	1	scaling lateral friction	
λ_{K_y}	1	scaling slip stiffness	
λ_{K_x}	1	scaling cornering stiffness	
Pure Slip Lateral			
P_{CY1}	1	shape factor C	chosen for convex model
P_{DY1}	1.0	friction at $F_{z_{nom}}$	fit at max. lateral acceleration
P_{DY2}	-0.065	load sensitivity	average value from [36]
P_{KY1}	-25	$K_{y\alpha}$ at $F_{z_{nom}}$	fit at low lateral accelerations
P_{KY2}	2	position of peak in $K_{y\alpha}$	recommended in[28]
P_{EY1}	-2	curvature at $F_{z_{nom}}$	fit at high lateral accelerations
P_{EY2}	0	curvature under load change	recommended in[28]
Pure Slip Longitudinal			
P_{CX1}	1	shape factor C	chosen for convex model
P_{DX1}	1.15	friction at $F_{z_{nom}}$	fit at max. braking manoeuvre
P_{DX2}	-0.065	load sensitivity	taken identical to lateral
P_{KX1}	19	$K_{x\kappa}$ at $F_{z_{nom}}$	fit for acceleration
P_{KX2}	0	position of peak in $K_{x\kappa}$	recommended in[28]
P_{EX1}	-1	curvature at $F_{z_{nom}}$	fit for max. (ABS) braking
P_{EX2}	0	curvature under load change	recommended in[28]
Pure Slip Aligning Moment			
Q_{CX1}	1	shape factor C pneumatic trail	example [28]
Q_{DZ1}	1.15	peak trail at $F_{z_{nom}}$	example [28]
Q_{DZ2}	-0.065	Variation of peak with load	example [28]
Q_{BX1}	19	Trail slope factor for trail at $F_{z_{nom}}$	example [28]
Q_{BX2}	0	Variation of slope Bpt with load	example [28]
Effective Rolling Radius			
$B_{R_{eff}}$	8.39	Low load stiffness	example [28]
$D_{R_{eff}}$	0.26	Peak value stiffness	example [28]
$F_{R_{eff}}$	0.0	High load stiffness	example [28]
C_z	2.1e5	vertical stiffness	fitted
Q_{RE0}	1.025	static scaling	fit at low speed dataset
Q_{V1}	0.0005	tyre radius increase with speed	fit at max. speed dataset
Q_{V2}	0.04	tyre stiffness increase with speed	example [28]

B.3 Tyre Property Plots

This section will illustrate the tyre characteristics for the selected parameters and show the influence of parameters on these characteristics.

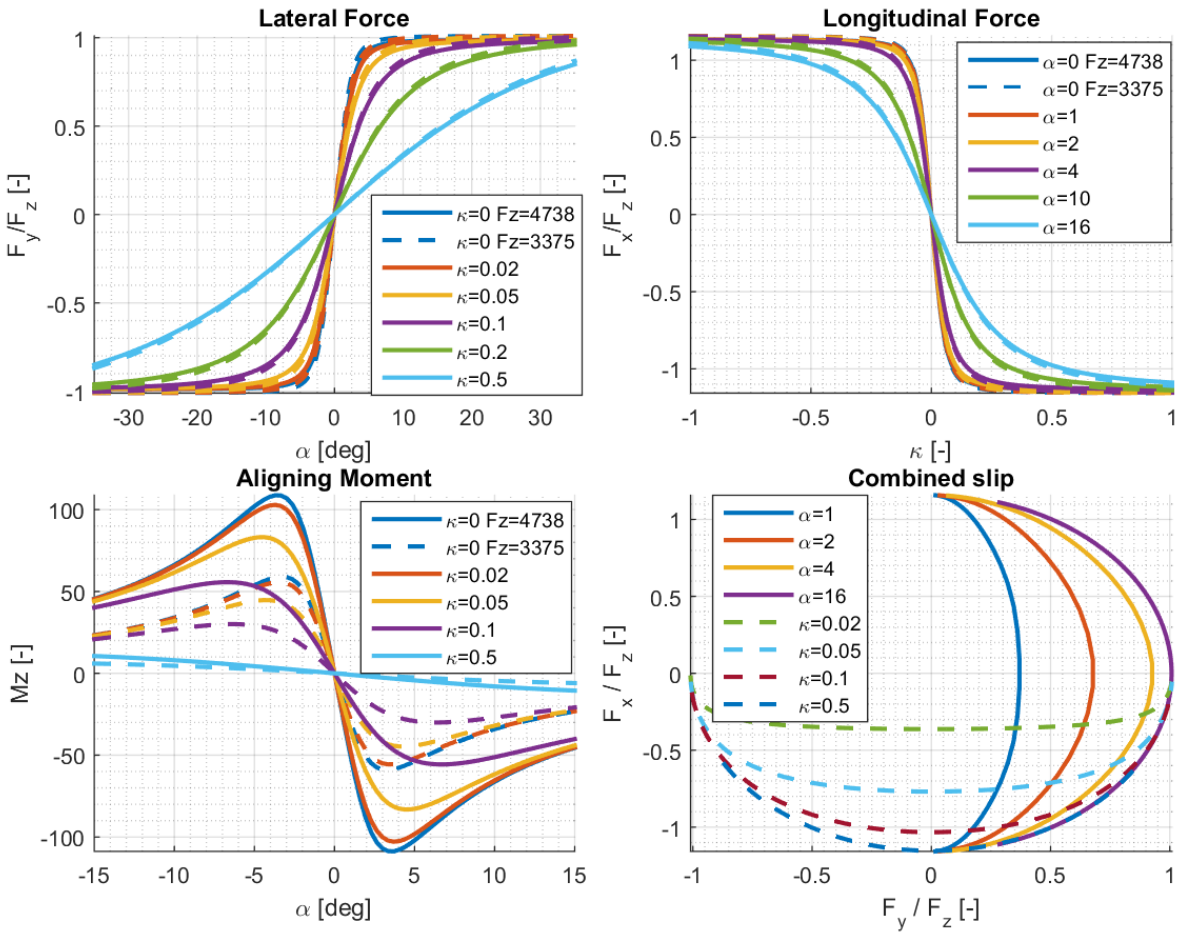


Figure B.1: Tyre characteristics for parameters given in Table B.1

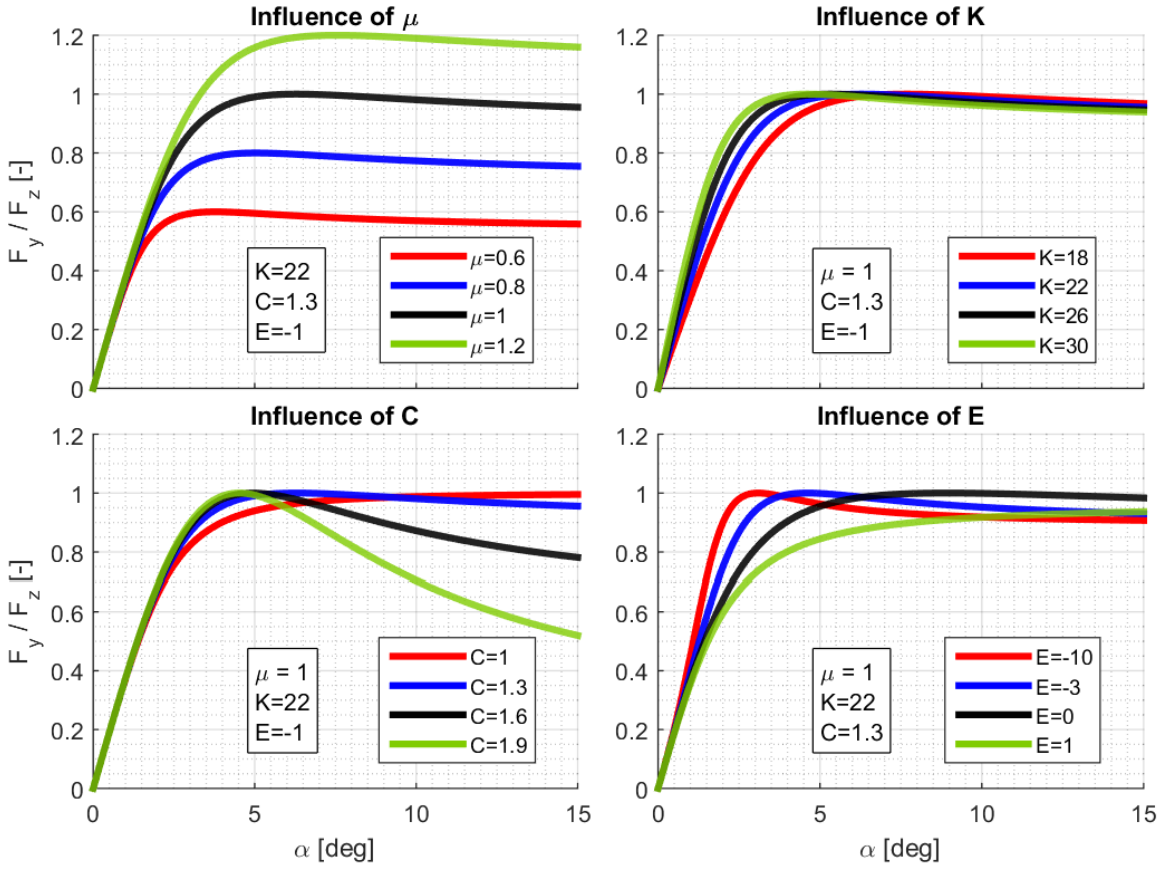


Figure B.2: Influence of parameters on characteristics of tyres

B.4 Relaxation Length Effects

This section demonstrates the effect of relaxation length, on the quality of the model outputs. Particularly in longitudinal direction. It can be seen in Figure B.3, that for low speeds, the noise on the wheelspeed signal has a significant impact on the longitudinal acceleration. Including the relaxation length solves this problem, as it functions as a speed dependent low pas filter on the slip velocity. The resulting reduction in output noise can be seen in Figure B.4.

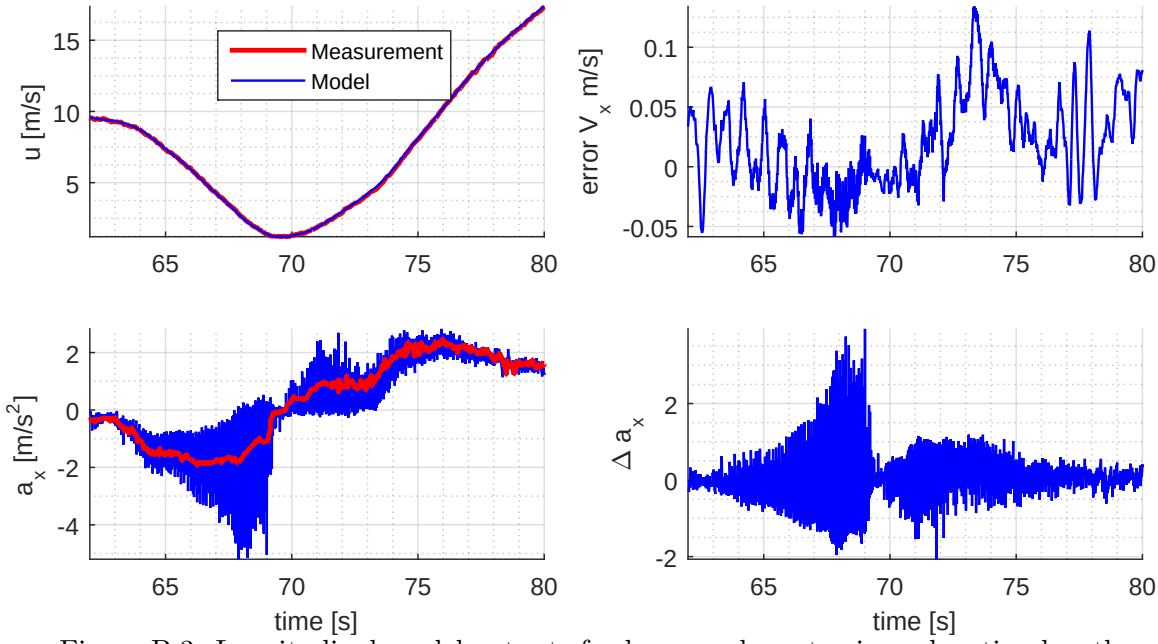


Figure B.3: Longitudinal model outputs for low speeds, not using relaxation length

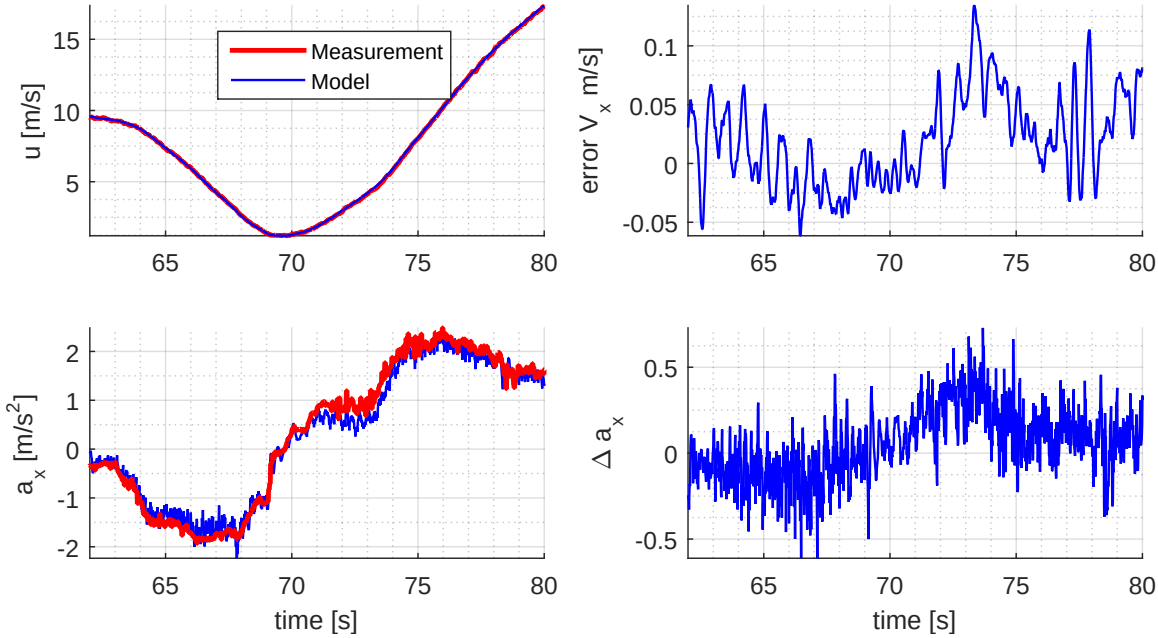


Figure B.4: Longitudinal model outputs for low speeds, using relaxation length of $\sigma_{rel_{x,y}} = 0.3m$

Appendix C

Estimator Tuning

Tuning of the parameters and settings used in the estimator are essential for performance and practical robustness of the estimator. To simplify implementation of the estimator when applied to a new vehicle, two flowcharts are presented in Figure C.1 and Figure C.2. These flowcharts describe a step-by-step tuning process and lists which kind of measurements should be used. It is assumed here that no prior knowledge of the tyres is available. When tyre parameters are available, it is evident that these parameters have the preference over manually tuned parameters. Note that when these tyre parameters are available the tuning process will instead by dominated by parameters related to the vehicle rather then the tyre. Also note that not all tyre parameters in Table B.1 are presented in the flowchart. These values are instead taken from literature or example values from elsewhere.

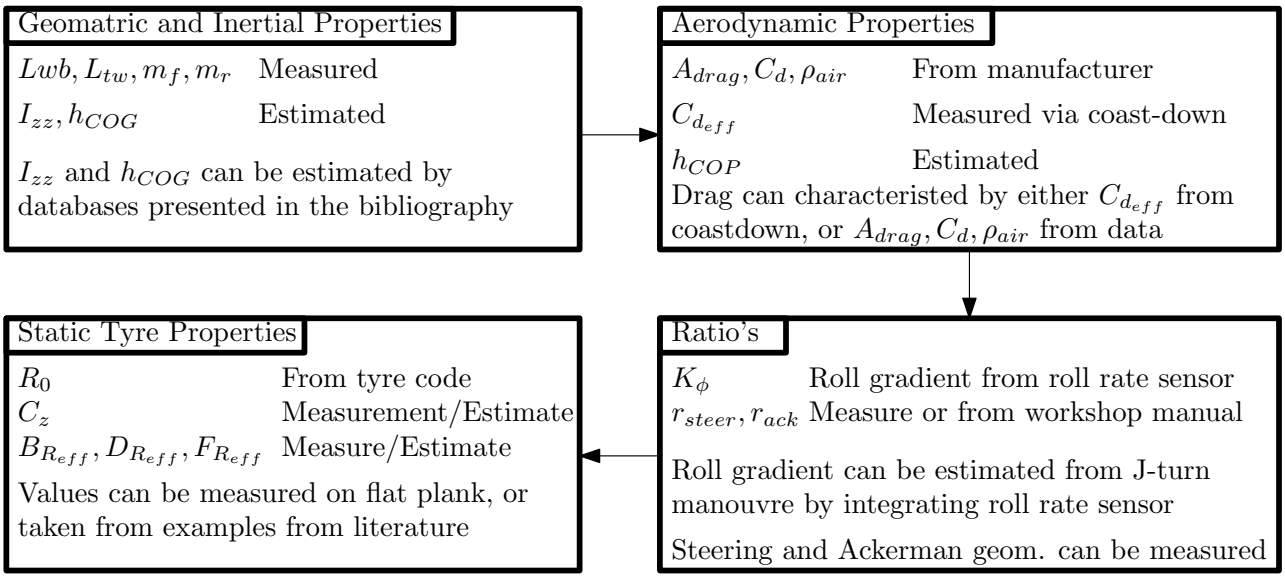


Figure C.1: Flowchart for tuning the estimator part 1

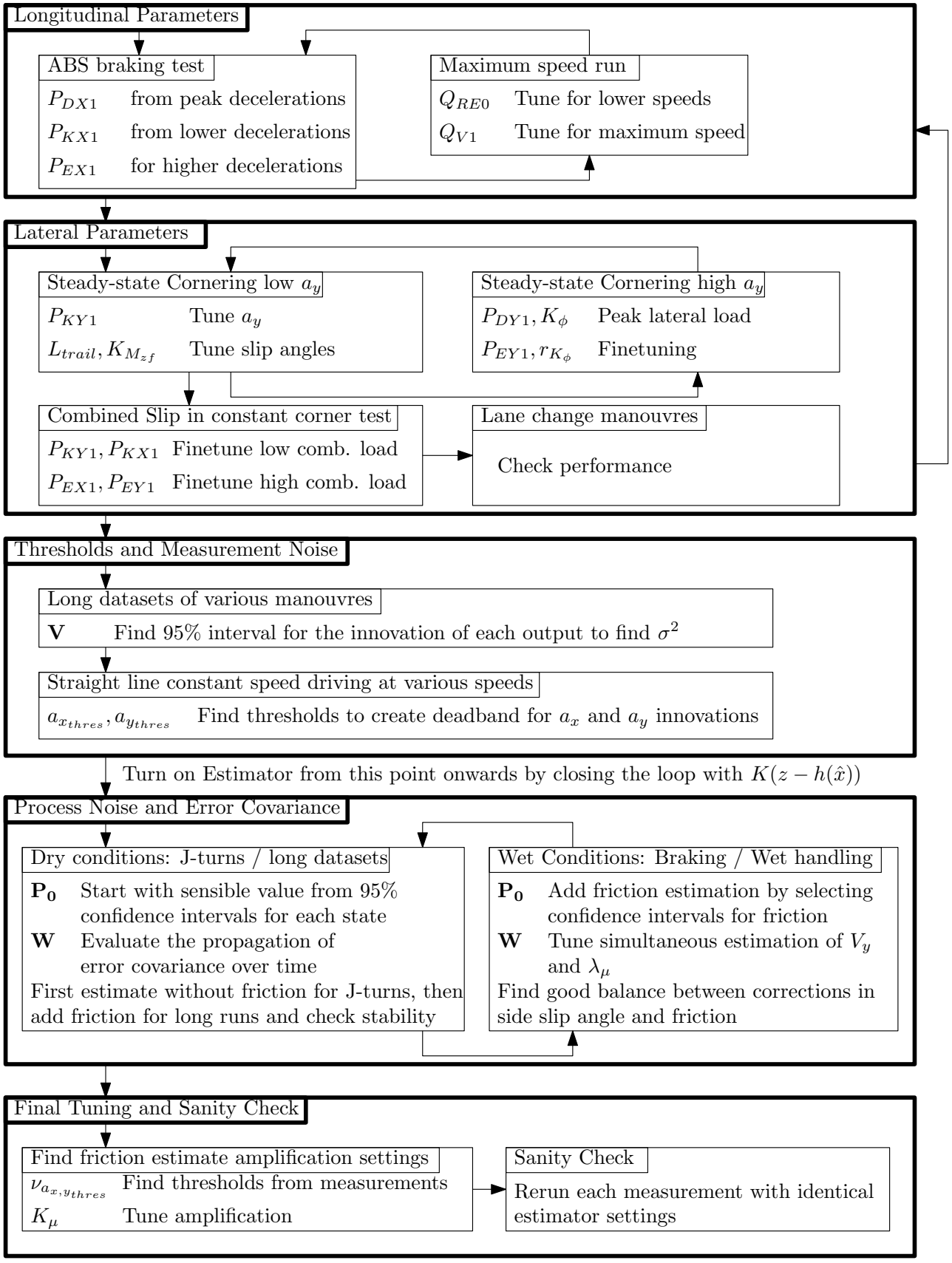


Figure C.2: Flowchart for tuning the estimator part 2

Appendix D

Test Set-up

D.1 Sensors

This section will describe the sensors that are used in the vehicle. An overview of the signals that are used from the vehicle can be found in Table D.1. Four wheelspeed signals are used in combination with the steering wheel angle sensor. The latter has a resolution of 1.7 deg. Acceleration and rotational velocities are taken from the XSens [40] IMU and are used for correction of the states. A Correvit [41] optical sensor is used as reference for the longitudinal and lateral velocity. Note that the optical sensor is not required for running the estimator and only functions as a reference to which to compare the estimator results with.

Table D.1: Sensors in the Toyota Prius in the Vehicle State Estimator

Signal	**Symbol**	**Source**	**Used for**
Wheelspeeds	$\omega_{1,2,3,4}$	Vehicle CAN-bus	Model Input
Steering angle	δ_{sw}	Vehicle CAN-bus	Model Input
longitudinal acc.	a_x	Xsens IMU	Correction
lateral acc.	a_y	Xsens IMU	Correction
yaw rate	r	Xsens IMU	Correction
roll rate	$\dot{\phi}$	Xsens IMU	Reference correction
longitudinal velocity	V_x	Corrsys	Reference
lateral velocity	V_y	Corrsys	Reference

D.1.1 Logging Frequency

The signals are logged with a frequency of 100 Hz. However, the values are only updated by the sensors at a frequency of 25Hz. This results in a 25 Hz signal, up-sampled to a frequency of 100 Hz with a zero order hold.

D.1.2 Acceleration Measurement Correction

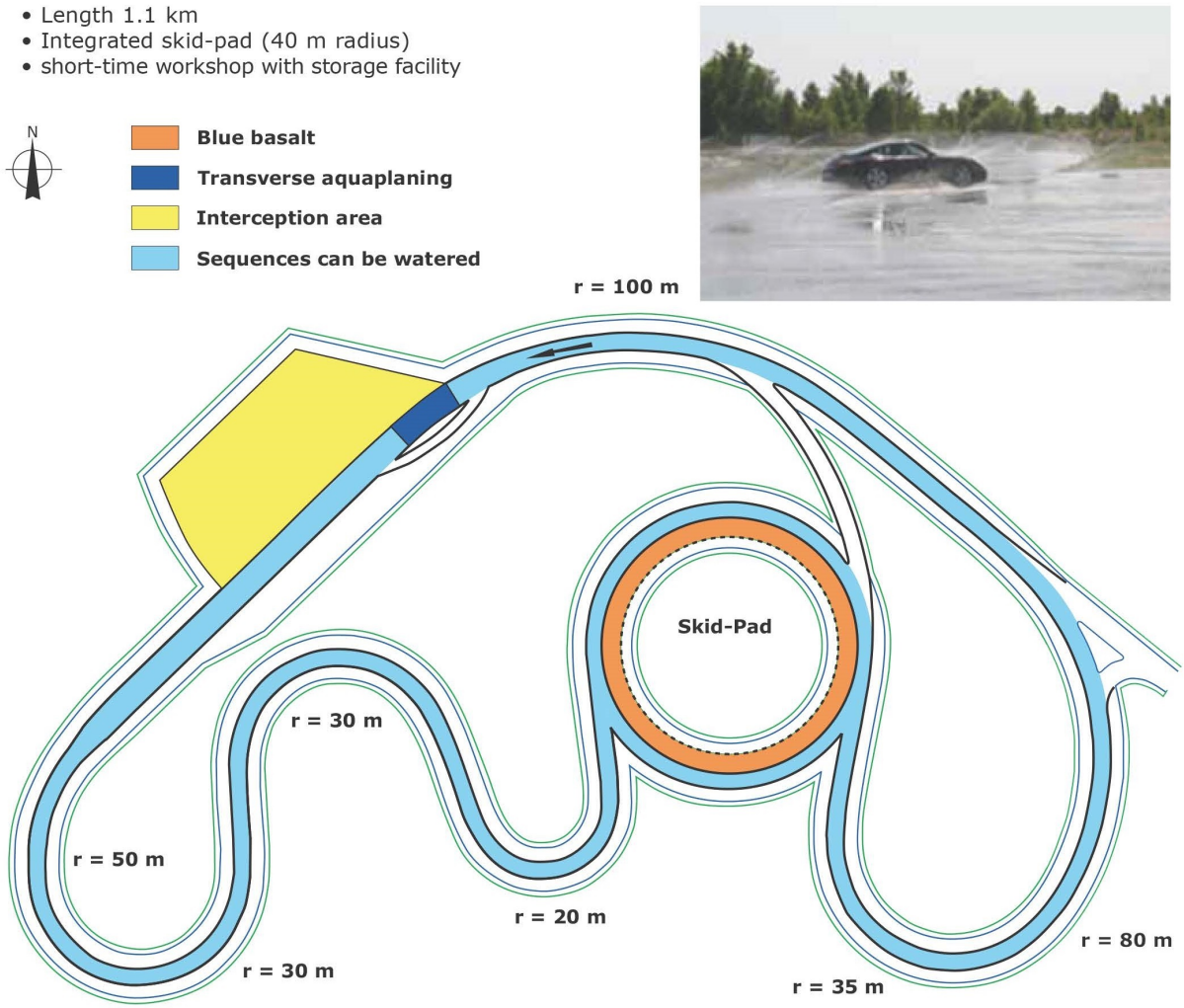
On banked or inclined roads or with roll or pitch angles of the body with respect to the suspension, the measured accelerations do not correspond with the in-plane accelerations. This is due to a component of gravity being measured. This can also be used to simultaneously estimate road bank angle and inclination. The additional component is found by:

$$\begin{bmatrix}\Delta a_x\\ \Delta a_y\\ \Delta a_z\end{bmatrix} = \begin{bmatrix}\mathrm{c}\psi\ \mathrm{c}\theta & \mathrm{s}\psi\ \mathrm{c}\theta & -\mathrm{s}\theta\\ \mathrm{c}\psi\ \mathrm{s}\theta\ \mathrm{s}\phi - \mathrm{s}\psi\ \mathrm{c}\phi & \mathrm{s}\psi\ \mathrm{s}\theta\ \mathrm{s}\phi + \mathrm{c}\psi\ \mathrm{c}\phi & \mathrm{c}\theta\ \mathrm{s}\phi\\ \mathrm{c}\psi\ \mathrm{s}\theta\ \mathrm{c}\phi + \mathrm{s}\psi\ \mathrm{s}\phi & \mathrm{s}\psi\ \mathrm{s}\theta\ \mathrm{c}\phi - \mathrm{c}\psi\ \mathrm{s}\phi & \mathrm{c}\theta\ \mathrm{c}\phi\end{bmatrix}\begin{bmatrix}0\\0\\-g\end{bmatrix} = \begin{bmatrix}g\sin\theta\\ -g\cos\theta\sin\phi\\ -g\cos\theta\cos\phi\end{bmatrix} \quad \text{(D.1)}$$

D.2 ATP

The tests are performed at the various tracks of the Automotive Testing Papenburg testtrack [37] as seen in Figure 6.1b. Aside from large open test tracks, where various manoeuvres can be tested easily, the ATP testtrack also features multiple sections where road friction can be adapted, ranging from wet asphalt to blue basalt. Although no true value is known as was the case in simulations, these tests can still be used to validate the ability of the estimator to adapt to the lower friction. Illustrations and information is all obtained from [37].

D.2.1 Wet Handling Course (NHK)

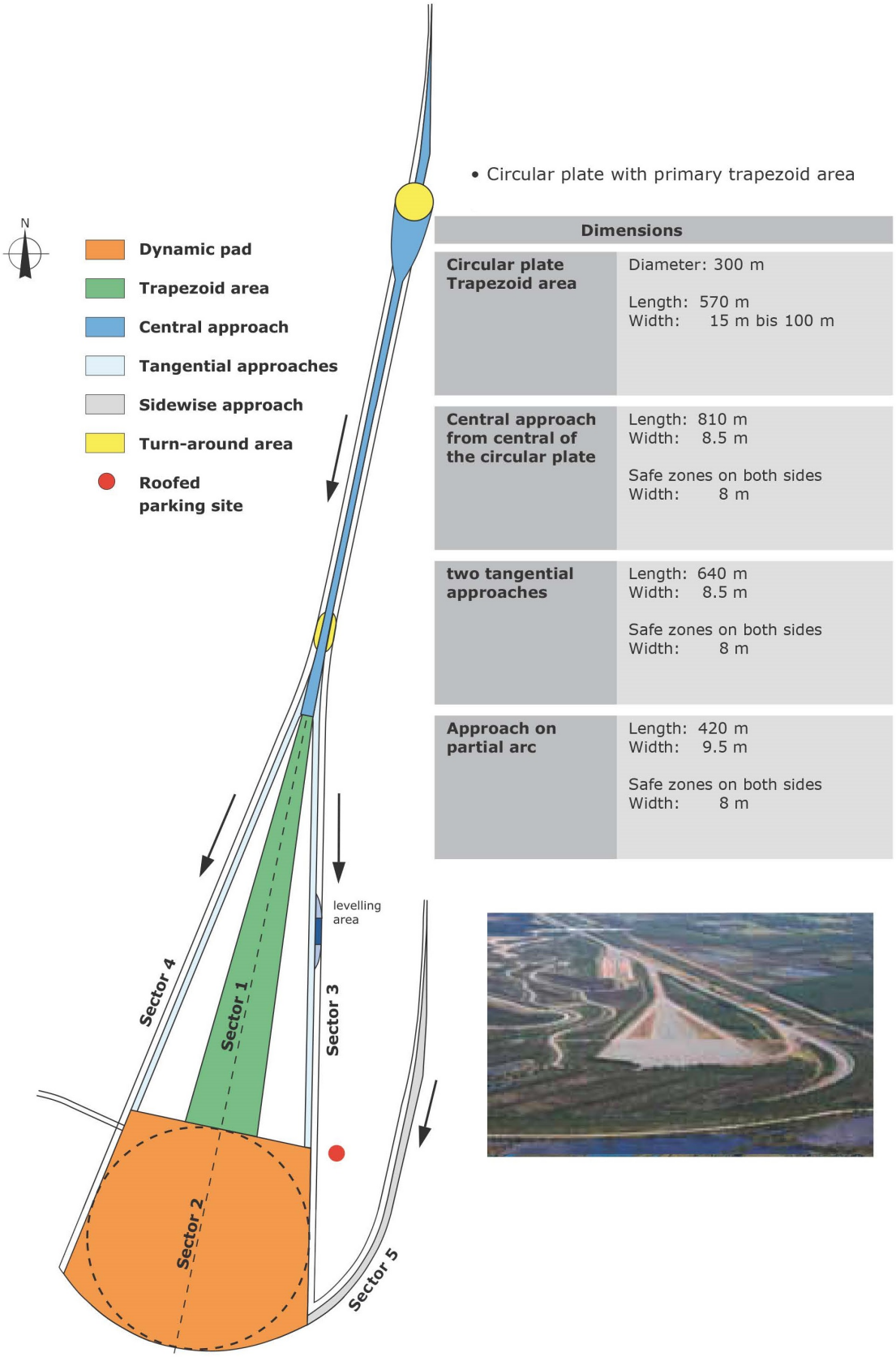


Wet handling/ aquaplaning with integrated skid-pad	
Total length	1.1 km
Lane width	5.0 m

Dimensions of skid-pad	
Asphalt	r = 40.0 m
Blue basalt	r = 35.0 m
Lane width	each 5.0 m

Figure D.1: NHK [37]

D.2.2 Vehicle Dynamics Area (FDY)

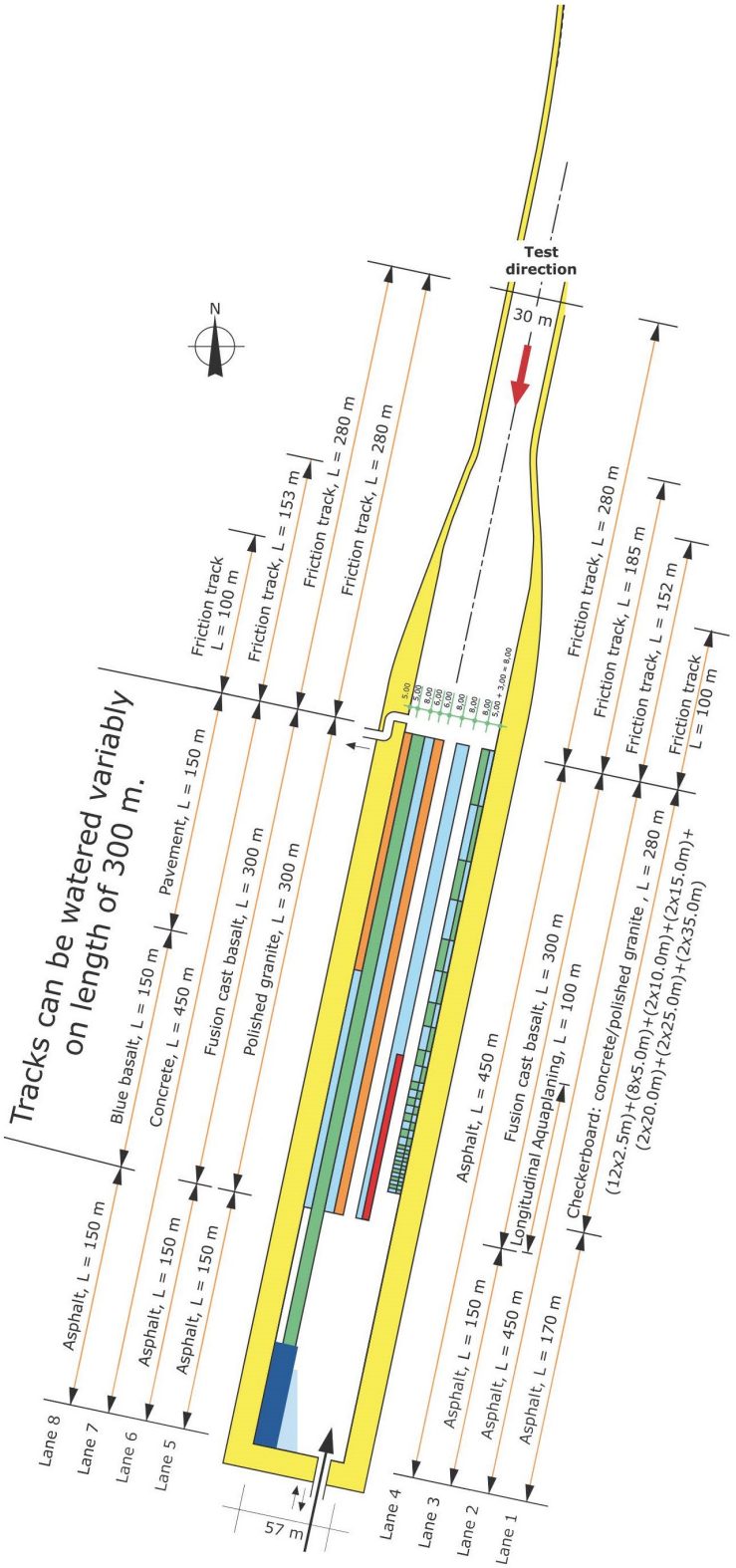


• Circular plate with primary trapezoid area

Dimensions	
Circular plate Trapezoid area	Diameter: 300 m Length: 570 m Width: 15 m bis 100 m
Central approach from central of the circular plate	Length: 810 m Width: 8.5 m Safe zones on both sides Width: 8 m
two tangential approaches	Length: 640 m Width: 8.5 m Safe zones on both sides Width: 8 m
Approach on partial arc	Length: 420 m Width: 9.5 m Safe zones on both sides Width: 8 m

Figure D.2: FDY [37]

D.2.3 Braking Tracks (BMK)



Lane	Surface	Length	Width
1	Checkerboard (concrete/polished granite)	280.0 m	8.0 m
	Asphalt	170.0 m	8.0 m
2	Asphalt	450.0 m	8.0 m
	Includes longgitudinal aquaplanning surface	100.0 m	4.0 m
3	Fusion cast basalt	300.0 m	8.0 m
	Temporarily construction testing with different surfaces		
	Asphalt	150.0 m	8.0 m
4	Asphalt	450.0 m	8.0 m
5	Polished granite	300.0 m	6.0 m
	Asphalt	150.0 m	6.0 m
6	Fusion cast basalt	300.0 m	6.0 m
	Asphalt	150.0 m	6.0 m
7	Concrete	450.0 m	8.0 m
8	Pavement	150.0 m	5.0 m
	Blue basalt	150.0 m	5.0 m
	Asphalt	150.0 m	5.0 m

- Braking tracks for cars and trucks

Figure D.3: BMK [37]

Appendix E

Additional Plots

This Appendix will provide some supportive plots that have been omitted in the report for the sake of brevity and readability. A small comparison between the non-convex and convex tyre model for actual experimental data is also included.

E.1 Innovation for Experiments

The innovation sequences of the experiments that are presented in Section 6.3 are presented here. This will give an impression of the accuracy of the modelled outputs under various conditions. Note that specifically in the case of longitudinal braking in Figure E.2, the difference in modelled and measured a_x is very large. This can be attributed to the lower real friction. These larger errors are used as described in 5.5.2.

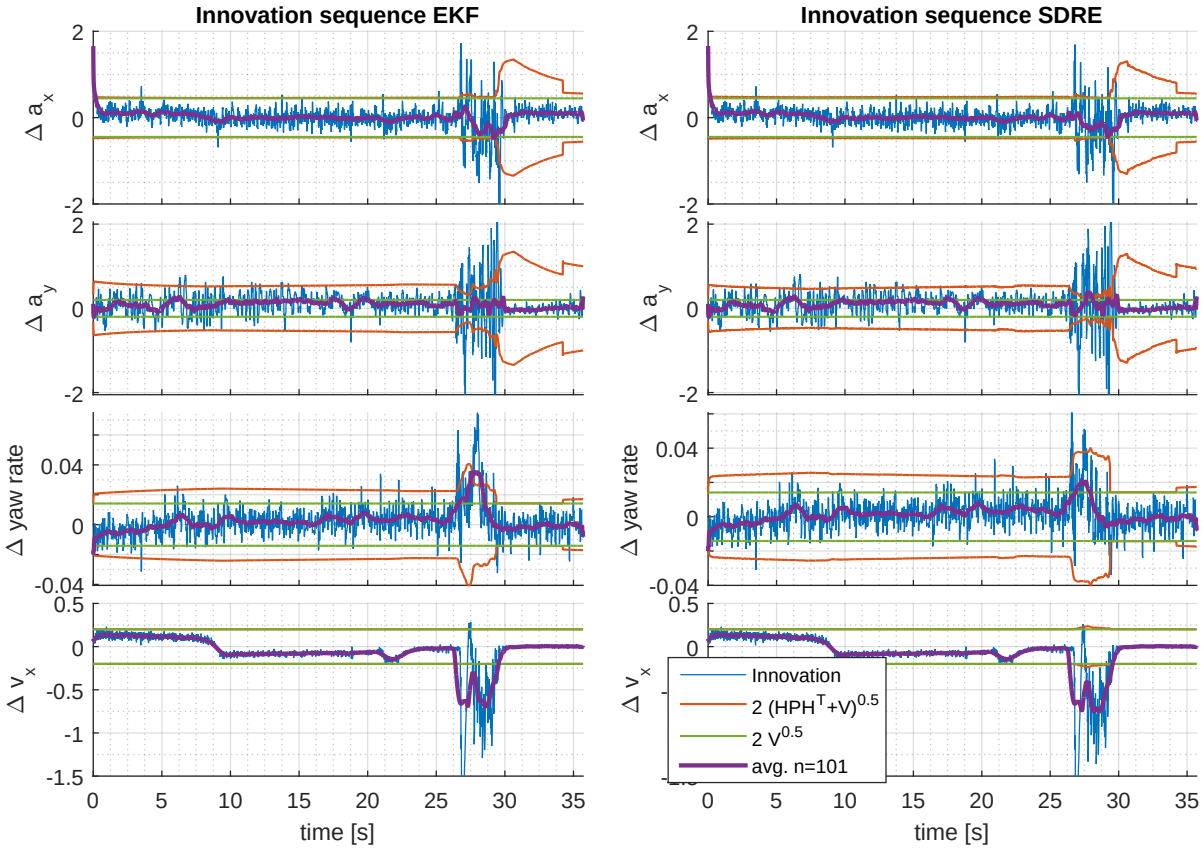


Figure E.1: Innovations corresponding to test in Figure 6.2

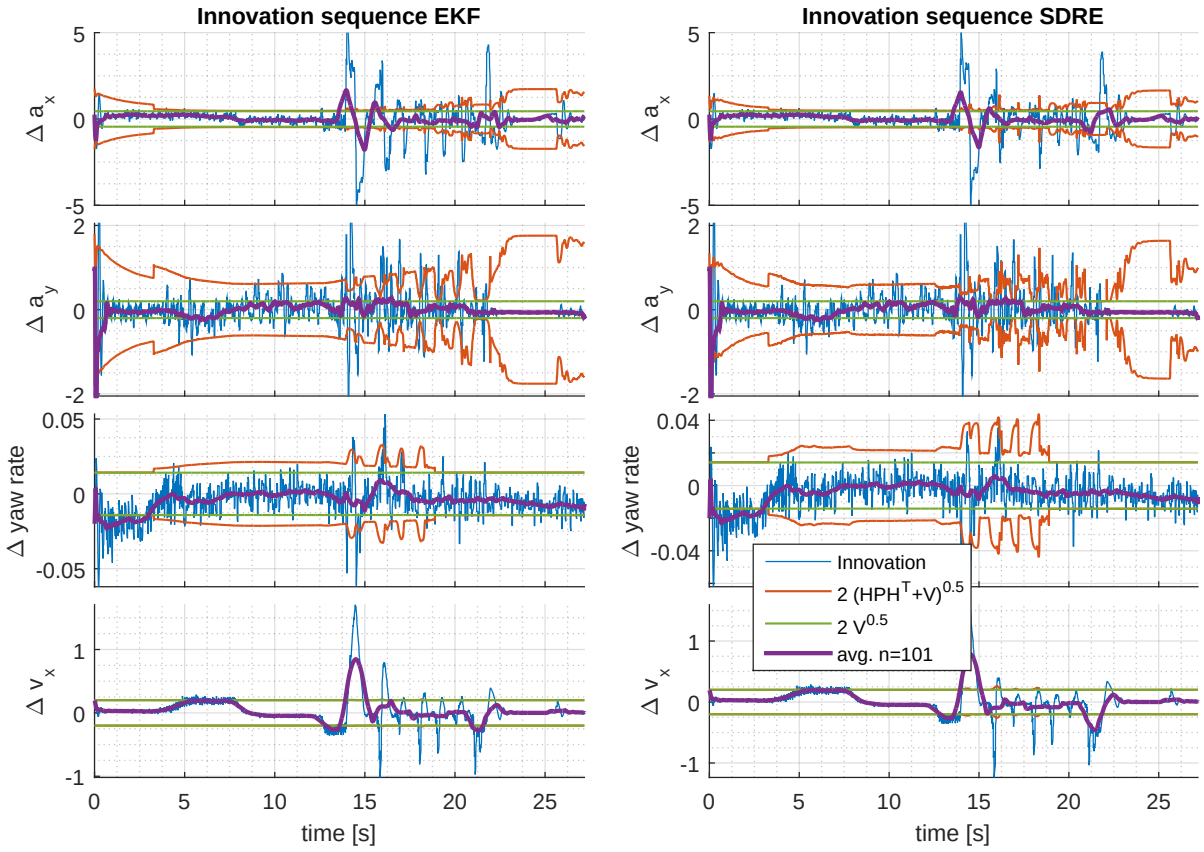


Figure E.2: Innovations corresponding to test in Figure 6.3

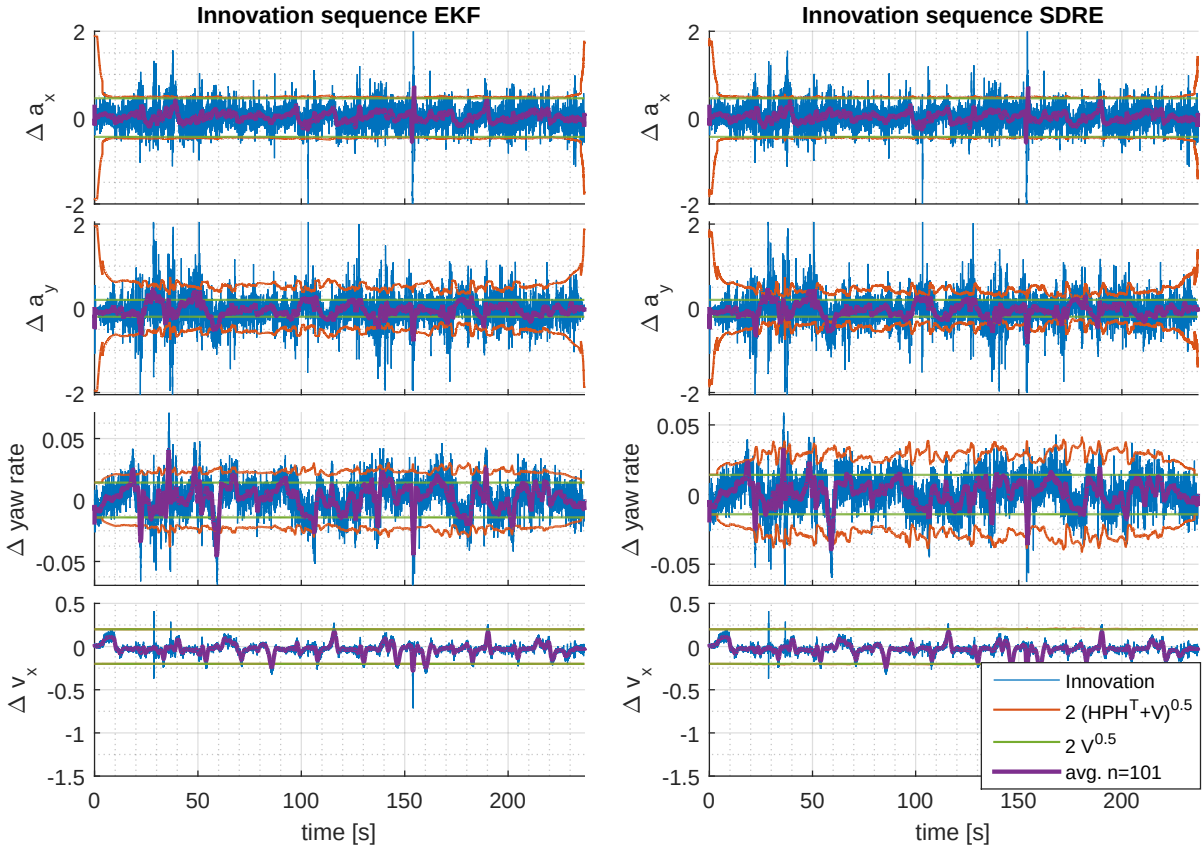


Figure E.3: Innovations corresponding to test in Figure 6.4

E.2 Model Convexity

Figure E.4 illustrates the state estimates also presented in Figure 6.4, but now featuring a peak in the lateral force characteristic of the vehicle model. This is achieved by setting the shape factor to $C_y = 1.6$. Both SDRE and EKF estimator are still capable of providing accurate estimates.

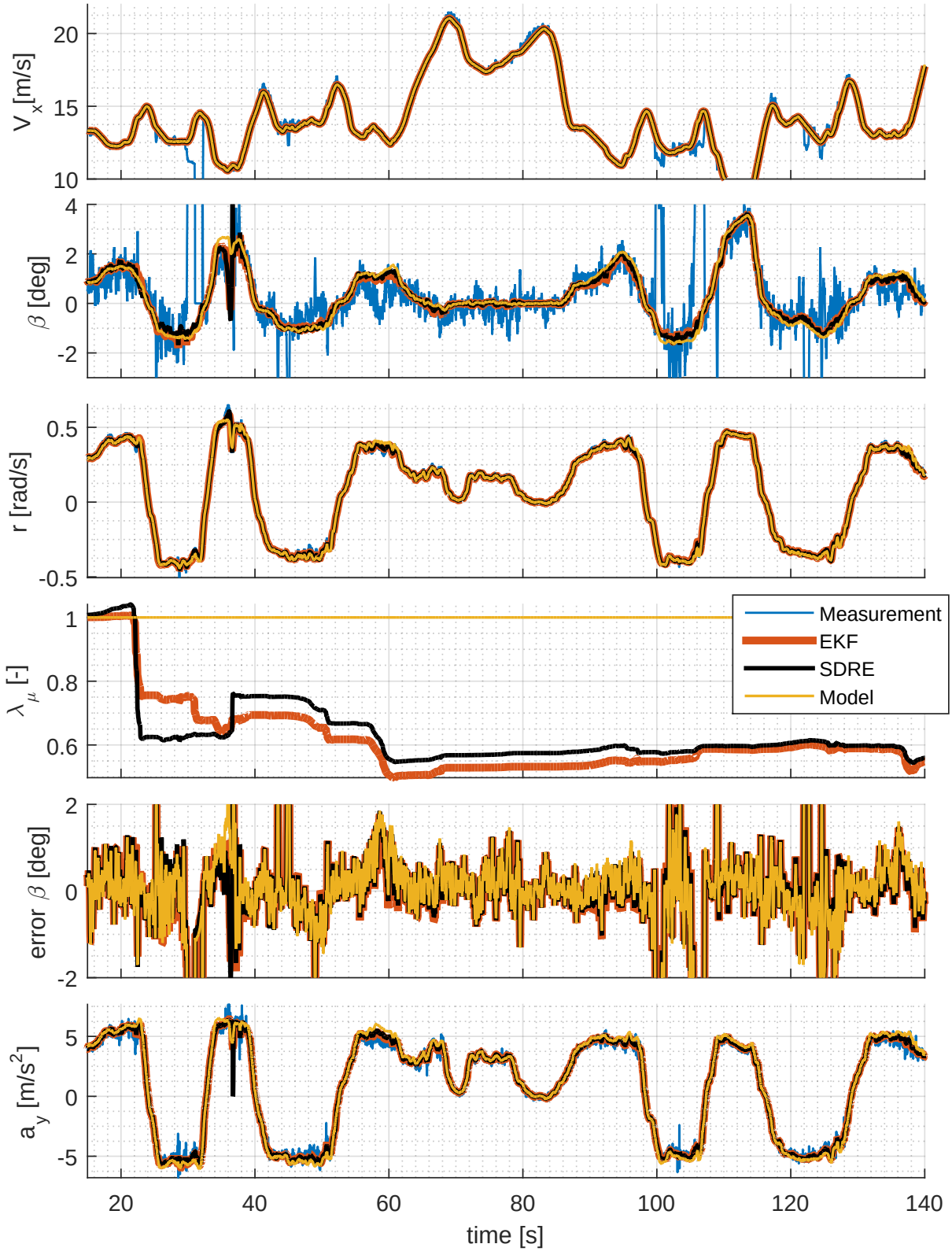


Figure E.4: Comparison of filter convergence, when the estimator model features a non-convex tyre model ($C_y = 1.6$)

In contrast to 2.3, the previous example shows fairly similar behaviour despite the non-convex tyre model. This is somewhat different from what was expected due to the very different convergence results presented in 2.3. However, the clear difference is the absence of dynamics in convergence study presented earlier. In the Vehicle State Estimation presented here, the dynamics strongly influence the trajectory of the estimates.

To visually represent this difference, a simplified bicycle model with only steering input is used to draw the vectorfield for a fixed steering angle and forward velocity in Figure E.5. The parameters used in this model again represent the Toyota Prius. Instead of steering compliance, the front axle cornering stiffness is reduced with respect to that of the rear axle. In the contour plot it can be identified that for a given yaw rate, multiple side slip velocities yield the same lateral acceleration. This is similar to the problem presented in 2.3. However, as can be seen from the vectorfield plot, the dynamics for these multiple side slip velocities very different. More importantly, the presence of these dynamics ensure that the estimates are not in equilibrium for a given lateral acceleration as was the case in 2.3. This explains why here the non-convex model does not yield significantly different behaviour between EKF and SDRE estimators.

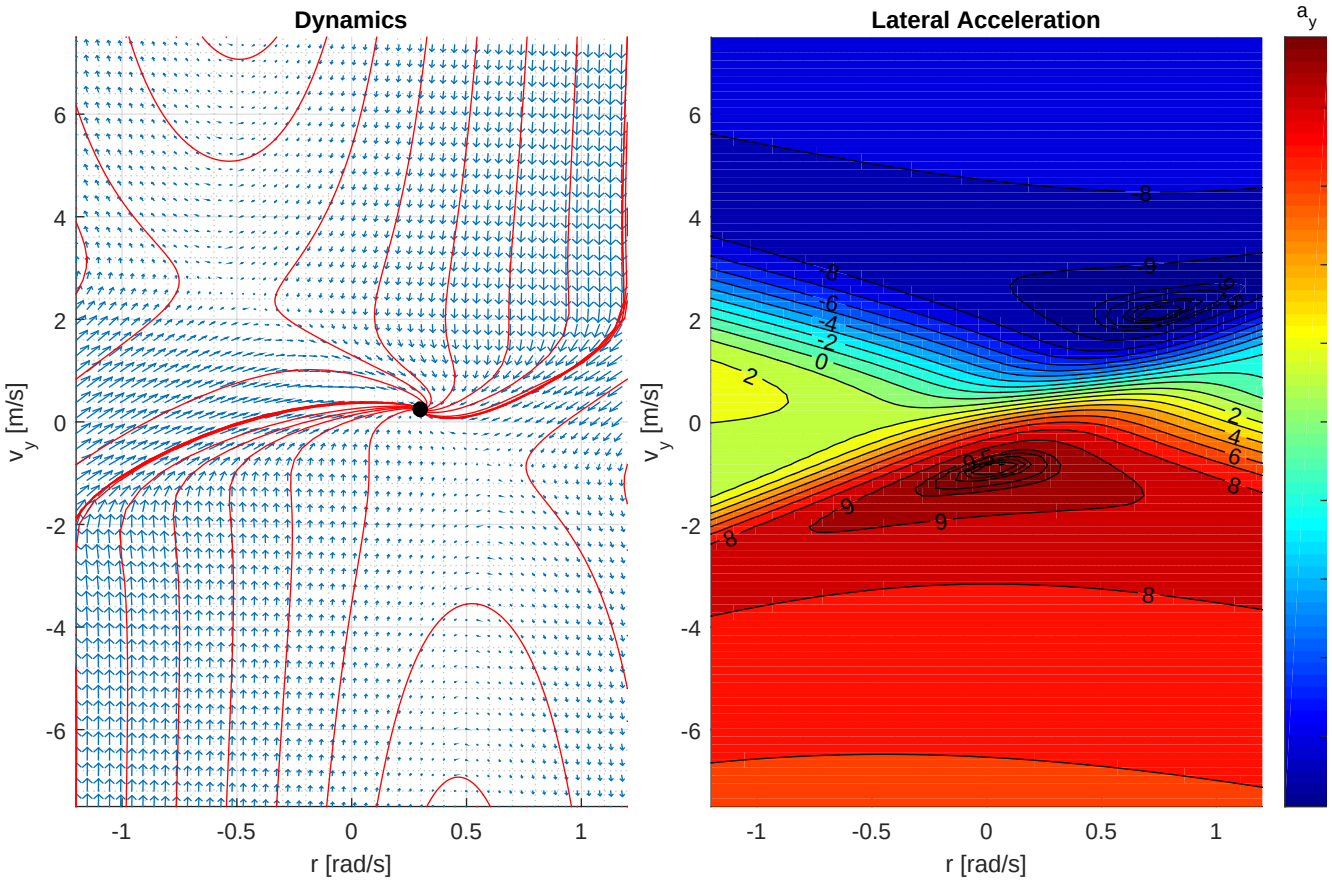


Figure E.5: Dynamics and Lateral acceleration for non-convex tyre model ($C_y = 1.6$), at forward velocity of $V_x = 15[m/s]$ and steering angle of $\delta_1 = 4[deg]$

Bibliography

[1] F. Yu, D.-F. Li, and D. Crolla, "Integrated vehicle dynamics control state - of - the art review," in *Vehicle Power and Propulsion Conference, 2008. VPPC'08. IEEE.* IEEE, 2008, pp. 1–6.

[2] B. Bonsen, R. Mansvelders, and E. Vermeer, "Integrated vehicle dynamics control using state dependent riccati equations," in *10th International Symposium on Advanced Vehicle Control, AVEC10, August 22-26, 2010, Loughborough, United Kingdom*, 2010.

[3] E. Wachter, "Limit handling collision avoidance system using state dependent riccati equation technique," Master's thesis, Chalmers University of Technology, 2016.

[4] J. R. Cloutier, C. N. D'Souza, and C. P. Mracek, "Nonlinear regulation and nonlinear h-infinity control via the state-dependent riccati equation technique. i- theory," in *International Conference on Nonlinear Problems in Aviation and Aerospace, 1 st, Daytona Beach, FL*, 1996, pp. 117–130.

[5] C. P. Mracek, J. Clontier, C. D'Souza *et al.*, "A new technique for nonlinear estimation," in *Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on.* IEEE, 1996, pp. 338–343.

[6] D. A. Haessig and B. Friedland, "A method for simultaneous state and parameter estimation in nonlinear systems," in *American Control Conference, 1997. Proceedings of the 1997*, vol. 2. IEEE, 1997, pp. 947–951.

[7] D. A. Haessig, "On-line state and parameter estimation in nonlinear systems," Ph.D. dissertation, New Jersey's Science & Technology University, 1999.

[8] D. A. Haessig and B. Friedland, "State dependent differential riccati equation for nonlinear estimation and control," in *Proceedings of the 15th IFAC Triennial World Congress*, 2002.

[9] T. Çimen, "State-dependent Riccati equation (SDRE) control: a survey."

[10] T. Çimen and A. O. Merttopçuoğlu, "Asymptotically optimal nonlinear filtering: Theory and examples with application to target state est," in *World Congress*, vol. 17, no. 1, 2008, pp. 8611–8617.

[11] A. Nemra and N. Aouf, "Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering," *Sensors Journal, IEEE*, vol. 10, no. 4, pp. 789–798, 2010.

[12] A. Berman, P. Zarchan, and B. Lewis, "Comparisons between the extended kalman filter and the state-dependent riccati estimator," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 5, pp. 1556–1567, 2014.

[13] C. P. Mracek and J. R. Cloutier, "Full envelope missile longitudinal autopilot design using the state-dependent riccati equation method," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 1997, pp. 1697–1705.

[14] A. K. Madhusudhanan, M. Corno, B. Bonsen, and E. Holweg, "Solving algebraic riccati equation real time for integrated vehicle dynamics control," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 3593–3598.

[15] H. Banks, B. Lewis, and H. Tran, "Nonlinear feedback controllers and compensators: a state-dependent riccati equation approach," *Computational Optimization and Applications*, vol. 37, no. 2, pp. 177–218, 2007.

[16] H. Beikzadeh and H. D. Taghirad, "Exponential nonlinear observer based on the differential state-dependent riccati equation," *International Journal of Automation and Computing*, vol. 9, no. 4, pp. 358–368, 2012.

[17] H. Durrant-Whyte, "Introduction to estimation and the kalman filter," January 2001.

[18] T. Karvonen, "Stability of linear and non-linear kalman filters," Master's thesis, University of Helsinki, 2014.

[19] C. Cruz and H. Nijmeijer, "Synchronization through extended kalman filtering," in *New Directions in nonlinear observer design*. Springer, 1999, pp. 469–490.

[20] H. K. Khalil and J. Grizzle, *Nonlinear systems*, 3rd ed. Pearson, 2014.

[21] S. Nazari and B. Shafai, "Robust sdc parameterization for a class of extended linearization systems," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 3742–3747.

[22] H. Nijmeijer and A. Van der Schaft, *Nonlinear dynamical control systems*. Springer Science & Business Media, 2013.

[23] G. Welch and G. Bishop, "An introduction to the kalman filter. 2006," *University of North Carolina: Chapel Hill, North Carolina, US*, 2006.

[24] R. Faragher *et al.*, "Understanding the basis of the kalman filter via a simple and intuitive derivation," *IEEE Signal processing magazine*, vol. 29, no. 5, pp. 128–132, 2012.

[25] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[26] E. Wan, R. Van Der Merwe *et al.*, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE, 2000, pp. 153–158.

[27] W. F. Milliken and D. L. Milliken, *Race car vehicle dynamics*. Society of Automotive Engineers Warrendale, 1995.

[28] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.

[29] R. Leenen, E. Vermeer, H. Schouten, J. Zuurbier, "Vehicle state estimation," March 2008, TNO internal document.

[30] M. C. Best, "A new empirical exponentialtyre model," *International Journal of Vehicle Design*, vol. 65, no. 2-3, pp. 241–255, 2014.

[31] M. Struijk, "Central differential control of an active four wheel drive vehicle," Master's thesis, University of Technology Delft, 2015.

[32] Toyota Motor Coorperation, "2012 prius product information," Tech. Rep., 2012.

[33] D. D. MacInnis, W. E. Cliff, and K. W. Ising, "A comparison of moment of inertia estimation techniques for vehicle dynamics simulation," SAE Technical Paper, Tech. Rep., 1997.

[34] G. J. Heydinger, R. A. Bixel, W. R. Garrott, M. Pyne, J. G. Howe, and D. A. Guenther, "Measured vehicle inertial parameters-nhtsa's data through november 1998," *SAE transactions*, vol. 108, no. 6; PART 2, pp. 2462–2485, 2000.

[35] Toyota Motor Coorperation, "2012 prius workshop manual," Tech. Rep., 2012.

[36] B. Niessen, S. Jansen, I. Besselink, A. Schmeitz, and H. Nijmeijer, "An enhanced generic single track vehicle model and its parameter identification for 15 different passenger cars," in *AVEC'12, The 11th International Symposium on Advanced Vehicle Control, September 9-12, 2012, Seoul, Korea*, 2012.

[37] "ATP Papenburg Testtrack," http://www.atp-papenburg.de/en/, Accessed: 2016-06-09.

[38] "Speedgoat, real-time simulation and testing," https://www.speedgoat.ch/, Accessed: 2016-06-14.

[39] T. Jansen, "Tyre modelling for state estimation and control purposes emphasizing viscoelasticity," Master's thesis, Eindhoven University of Technology, 2016.

[40] "XSens, Inertial Measurement Unit," https://www.xsens.com/products/, Accessed: 2016-06-12.

[41] "Kistler, correvit ®2-axis optical sensor," https://www.kistler.com/nl/en/applications/automotive-research-test/vehicle-dynamics-durability/dynamics-testing/products/, Accessed: 2016-06-12.