POLITECNICO DI TORINO

Master's Degree
in Mechatronic Engineering

Master's Degree Thesis

Heteroscedastic noise estimation in Kalman filtering applied to road geometry estimation

Supervisor
Prof. Alessandro Rizzo

Candidate
Serena De Vito

External Supervisor
Sebastian Inderst

July 2020

Abstract

Road geometry estimation is essential for self-driving vehicles and modern advanced driver-assistance systems (ADAS). State-of-the-art techniques utilize a Kalman filter to perform road geometry estimation. A common assumption in these Kalman filters is that the process- and measurement noise covariances are constant over time. However, both sensor performance and process dynamics may change over time in real-world applications. Sensor performance may be affected by environmental factors such as rain and lighting conditions and process dynamics may depend on the road type. Noise processes like these with a feature dependent covariance are known as heteroscedastic noise. By estimating the heteroscedastic process- and measurement noise covariances more accurately both the filter performance and state uncertainty estimation may improve. Road geometry estimation is an especially interesting application in which to apply heteroscedastic noise estimation as there are several factors which intuitively should affect the process- and measurement noise.

In this thesis a framework for heteroscedastic noise estimation in Kalman filtering applied to road geometry estimation is presented. The framework consists of two parts; a feature selection part and a heteroscedastic noise model. This noise model is constructed offline based on data set of ground truth state vector data. Two different state-of-the-art approaches for heteroscedastic noise modeling, a parametric approach and an approach based on a deep neural network, are evaluated as to determine if they are suitable for the application of road geometry estimation. Furthermore, a straightforward approach that models heteroscedastic noise by dividing the features into discrete cases is studied.

The noise models are quantitatively evaluated using a likelihood measure and root mean square error of the road geometry estimation. The results show that heteroscedastic noise estimation may improve both filter performance and estimation uncertainty consistency in the application of road geometry estimation.

Keywords: Road geometry estimation, Kalman filter, Heteroscedastic noise estimation, Noise modeling, Self-driving vehicles, Advanced driver assistance systems.

Acknowledgements

First of all, I would like to thank my academic supervisor, professor Alessandro Rizzo, for his guidance and prompt advise in the past months. I would also like to thank everyone at Zenuity, and in particular my supervisor Sebastian Inderst and team Kalman for helping me throughout this project and making me feel welcomed during my time in Sweden. I would like to thank my friends for these last epic years spent together. In particular, I would like to thank Rebecca for never getting tired of hearing my stories, Martina for inspiring me every day, and Maria for being my desk mate from day one. Last but not least, I would like to thank my family for leaving me the freedom to pursue my dreams and standing by me in every step of the way.

Serena De Vito, July 2020

Heteroscedastic noise estimation in Kalman filtering applied to road geometry estimation

Serena De Vito[1]
Rickard Persson[2]
(1) Department of Electronics and Telecommunications, Politecnico di Torino
(2) Department of Electrical Engineering, Chalmers University of Technology

Road geometry estimation is a vital part of several ADAS features. Road geometry for ADAS systems is typically done by using lane markers, target vehicles and map data. ADAS systems are of great interest for Zenuity AB. The focus of this project is the need to better study and define the uncertainty of the sensor data in order to adapt the noise correctly to the filters and therefore be able to correctly estimate the uncertainty of the road geometry filter. This project was carried out in the Göteborg office of Zenuity in collaboration with Team Kalman and under the guidance of the company supervisor Sebastian Inderst as part of the Master Thesis Program 2020.

Structure of the thesis

The structure of the thesis report is as follows.
Chapter 1 introduces the road geometry uncertainty estimation problem and proposed solution and thesis objective.
Chapter 2 introduces general theory relevant to understand both the methods used and the thesis background.
Chapters 3 and 4 present the work carried out by candidate Serena De Vito. Chapter 3 describes a preliminary data analysis and feature selection methods used. Chapter 4 presents the Discrete Covariance Estimation method. Chapter 5 presents two parametric approaches to the covariance estimation problem developed by Rickard Persson, the Parametric Covariance Estimation and the Deep Covariance Estimation. In these chapters theory specific to each part or method is initially presented followed by the concepts of the methods.
Chapters 6 and 7 present the results and corresponding discussion for each of the methods introduced.
Lastly, conclusions of the thesis work and future work suggestions are presented in Chapter 8.

Contents

List of Figures

List of Tables

1

Introduction

Modern vehicles are today becoming more and more sophisticated with regards to autonomy and sensing their environment. Vehicles have advanced driver-assistance systems (ADAS) which aid the driver by making driving both safer and more convenient. These systems include functionalities such as adaptive cruise control, automatic braking systems and collision avoidance systems. Furthermore, academia and industry are working towards making fully self-driving vehicles a reality which may further improve safety and sustainability within transportation. One of the reasons that self-driving vehicles may improve traffic safety is that human errors e.g. driving during fatigue, driving under influence, speeding and careless driving are common reasons for traffic accidents [1]. These causes could be completely avoided by the use of self-driving vehicles. With regards to sustainability self-driving vehicles enable a number of different strategies which may be used to reduce emissions and fuel consumption. These strategies include platooning [2], energy efficient control systems [3] and traffic flow control [4].

Road geometry estimation may be defined as in [5] which describes it as the problem of estimating the shape of the middle of a host vehicle's lane. Road geometry information is crucial for both ADAS and self-driving vehicles. ADAS require knowledge of road geometry to recognize if intervention is necessary e.g. if the car is diverging from the driving lane and road geometry estimation is fundamental for self-driving vehicles as the vehicle needs a reference path to follow.

Vehicles are equipped with a multitude of different sensors to sense the environment. These sensors may include radar, lidar, cameras, GPS and inertial measurement units (IMUs) among others. The sensors provide the host vehicle with noisy observations of e.g. lane markings and other road vehicles from which the road geometry may be derived. To fuse these different sensor measurements into a single estimate of the road geometry a common approach is to use a Gaussian filter which is evident from the work in [6][7][8][5]. The Kalman filter is one example of a Gaussian filter used for road geometry estimation. The Kalman filter relies on knowledge of sensor- and process dynamic uncertainties to produce accurate estimates of the road geometry and the corresponding estimation uncertainty. If the sensor- and process dynamic uncertainties are not described accurately it may result in a suboptimal filter or even cause the filter to diverge [9].

A criticism of the objective of noise estimation in mathematical models may be that one should aim to reduce the errors instead of trying to capture their variation. However, as mentioned in [10], mathematical models are only approximations of reality and as such they will not describe reality perfectly. Furthermore, the cost of estimating the error covariances may be lower compared to using a more complex model which would result in a reduction of the errors. Because of these reasons the objective of covariance estimation is a valuable objective to pursue.

As self-driving vehicles and ADAS are safety critical systems it is very important for these systems to be aware of their uncertainty. There are several different sources of uncertainty in road geometry estimation. The sensor measurements have intrinsic uncertainties and the estimates produced by the Kalman filter have further uncertainties following from modeling errors, assumptions and simplifications. By being aware of these uncertainties the control system may decide to control the vehicle more conservatively when the uncertainty is large as to not rely on an uncertain estimate. The Kalman filter uncertainty estimate is heavily reliant on knowledge of sensor- and process dynamic uncertainties in the same way as the filter estimate. It is therefore crucial to describe these uncertainties accurately such that the vehicle may be aware of the uncertainty and take safe actions.

1.1 Project objective

The objective of this thesis is to construct models that accurately estimate feature dependent measurement- and process noise covariances in a Kalman filter used for road geometry estimation. As a result both road geometry estimation performance and the accuracy of the estimation uncertainty may possibly improve. The models are to be used online but are constructed offline based on a data set of ground truth state vector data.

The estimation uncertainty in road geometry estimation may depend on a multitude of different factors. Sensor performance may be affected by environmental factors such as rain and light conditions and road geometry dynamics may depend on the type of road being driven. Measurements describing these factors are called features and may be informative for estimating the measurement- and process noise covariances. These kind of features and possibly other useful features are thus the inputs of the models constructed in this thesis. Furthermore, as the models predict measurement- and process noise covariances the model outputs consist of covariance matrices. An illustration of the model structure is given in Figure 1.1.

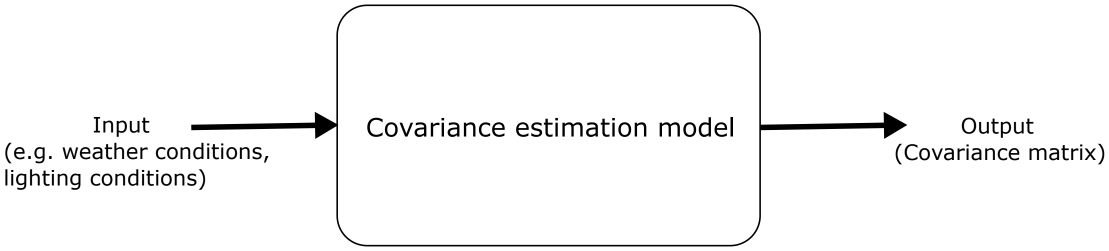


Figure 1.1: Illustration of the overall model structure with input and output.

A more rigorous mathematical description of the objective may also be formulated. Denoting the model input feature vector at time instance k as $\mathbf{z}_k \in \mathbb{R}^{n_z}$, where n_z is the number of input features, the overall objective is to model the functions $f(\mathbf{z}_k)$ and $g(\mathbf{z}_k)$ that map input features to the accurate measurement noise covariance R_k and process noise covariance Q_k respectively at each time instance k. The objective may thus be summarized as modeling f and g such that

$$\begin{aligned} R_k &= f(\mathbf{z}_k) \quad \forall k \\ Q_k &= g(\mathbf{z}_k) \quad \forall k. \end{aligned}$$

To evaluate how well the models estimate the true measurement- and process noise covariances two different performance measures are used. The first measure is based on the likelihood of the test data given the estimated covariances and it is a direct performance measure of the noise models. The second measure is related to the road geometry estimation performance, as more accurate measurement- and process noise covariances should result in a better filter performance. Exact definitions of the performance measures are given later in Chapter 6.

1.2 Related work

Filtering improvement methods known as adaptive Kalman filters have been developed with the aim of helping the filter to accurately estimate model parameters in the presence of model errors. Adaptive Kalman filters largely use the measurement noise process v_k, also known as measurement innvoation, defined in (2.3), to define a measure of the optimality of the filter. Mehra developed an adaptive Kalman filter to estimate the process and measurement noise covariance matrices Q and R online based on the measurement innovations [11]. The optimality of the filter is checked carrying out a statistical test based on the innovation properties of an optimal filter. Mohamed and Schwarz developed an adaptive Kalman filter based on the maximum-likelihood approach to make decisions on the proper choice of the filter gain factors, and used successfully in applications such as inertial navigation system (INS) and global positioning system (GPS) [12]. However, this technique only works well in the presence of noise properties that vary slowly and smoothly over time. The drawback of relying on the adaptive Kalman filter is a more complex algorithm, which is acceptable in cases that require a high accuracy. Moreover, the adaptive Kalman filter tries to reduce or bound the errors by adapting the model to real data, which results in a reactive algorithm, rather than a predictive algorithm, with high reaction time.

A number of covariance estimation methods for specific applications have been developed. Olson and Censi both address the localization and scan matching problems. Censi developed a covariance estimation method for the Iterative Closest/Corresponding Point algorithm (ICP), based on the analysis of the error function being minimized [13]. In his paper [14], Olson examined a family of probabilistically-motivated algorithms to calculate the alignment cost function at points around the global minima. Pupilli and Calway described an innovative visual Simultaneous Localization And Mapping (SLAM) algorithm based on particle filtering [15]. The coupling between the unscented Kalman filter (UKF) and the particle filter has proved to give the system resilience to unpredictable, erratic motions. These domain-specific covariance estimation methods are highly specialized, therefore they do not generalize well to other applications.

Covariance matrices are required to be positive definite matrices, which makes regression complicated. To overcome this problem, parametric covariance decomposition methods are used in order to regularize the sample covariance of high-dimensional data [16]. A popular form of decomposition is the modified Cholesky decomposition, which was first used by Pourahmadi in [17]. The great advantage of this decomposition is that it has only positive value constraints, however the parameter fitting process becomes complicated due to the large number of model parameters used in a parametric approach. In this thesis we refer to a straightforward parametric method described by Hu and Kantor, in their paper [16]. A more detailed discussion can be found in Section 5.2.

Recent works also provide with nonparametric techniques. In [18], Kersting et al. present a Gaussian process (GP) approach to regression to estimate varying noise variances, in the presence of input-dependent noise. The technique presented approximate the posterior noise variance using a most likely noise approach. A very popular nonparamteric method is the Covariance Estimation and Learning through Likelihood Optimization (CELLO) method presented by Vega-Brown et al. in [19]. The nearest-neighbor algorithm used provides an extension to the standard Gaussian measurement model with constant covariance, relying on on-line state estimation. This nonparametric kernel estimation technique approximates the sensor noise as the empirical covariance of neighbors of training data in a given feature space. As a consequence, this technique suffers from increasing computational complexity and memory requirements with the increasing of the size of the training data set, thus it is not well suited for large training data sets.

Among recent works, several methods have been dedicated to the direct learning of neural network models in probabilistic filters. In [20], Coskun et al. propose to learn dynamic representations of the motion and noise models from data using short-term memory. This approach finds representations that derive from all previous observations and states. In [21], Jonschkowski et al. learn prediction and measurement models using a differentiable particle filter (DPF). This method proves the advantage of combining end-to-end learning with algorithmic priors: the first optimizes model performance, while the second enables explainability and regularizes learning. In their paper, Kendall and Gal predict the variance of a deep neural network by modelling epistemic, i.e., deriving from the model, vs. aleatoric, i.e., inherent in the observations, uncertainty in Bayesian deep learning models [22]. In this thesis, we refer to an interesting approach to deep learning methods, given by the Deep Inference for Covariance Estimation (DICE) method developed by Liu et al. [23]. Further details on this approach are reported in Chapter 5.3.

1.3 Proposed solution and contributions

There exist state-of-the-art methods which aim to estimate covariances in Gaussian noise models in the general context of Gaussian filters. In this thesis some of these state-of-the-art methods are either modified to better fit the problem of road geometry estimation or applied directly to evaluate how useful they are for this specific application. A straightforward approach denoted Discrete Covariance Estimation (DCE) is also studied and evaluated.

Ultimately the contribution of this thesis is a framework for heteroscedastic noise estimation in Kalman filtering applied to road geometry estimation. That the noise is heteroscedastic means that the variance of the noise process is not constant and may therefore depend on some independent variable. The framework covers the workflow from selecting useful features to constructing the heteroscedastic noise models which provide the heteroscedastic covariance estimates. Regarding the heteroscedastic noise models our proposed solution is to model the noise processes in the Kalman filter model equations as zero mean Guassians with feature dependent covariances. These feature dependent covariances are estimated using models with learnable parameters where the models map from input features to covariance matrices. The parameters are learned offline based on a data set of ground truth state vector data. Additionally, interesting features are selected from a set of candidate features using feature selection methods.

Several parts of this thesis are to the best of our knowledge novel. Previous works apply the heteroscedastic noise models to measurement models only while in this thesis we also model the process noise covariance. This thesis aims to find heteroscedastic noise estimation models useful specifically for the application of road geometry estimation. We propose in Section 3.1.4 a feature selection method for continuous features in the specific case of heteroscedasatic covariance estimation models when one has access to samples of the random variable instead of the actual heteroscedastic covariances. In Section 5.1.3 we derive an argument as to why regularization is important in the objective function used in some of the state-of-the-art methods [10][23].

1.4 Reference system

At the start of the thesis a reference system for road geometry estimation was available. It uses a Kalman filter to produce road geometry estimates. Additionally, a heuristic method to estimate the measurement- and process noise covariances was available at the start of the thesis. The heuristic method consists of feature dependent noise model matrices and covariance matrices tuned based on filter performance. The heuristic method will be referred to as the baseline and is used to benchmark the proposed solutions within this thesis. The baseline and the covariance estimation methods evaluated in this thesis are applied to the Kalman filter reference system as to obtain comparable results.

The reference Kalman filter uses one prediction step and three different update steps. The prediction step may further be divided into two parts; a road prediction and an object prediction. The update steps are based on measurements of lane markings, positions of surrounding vehicles and headings of surrounding vehicles. The measurements in these update steps are obtained from camera and radar sensors. The performance of both camera and radar sensors are potentially heavily dependent on factors common during driving such as rain, darkness and motion blur from moving at high speeds. These performance dependencies make road geometry estimation a suitable candidate for heteroscedastic noise estimation.

To summarize the reference system has two process models and three measurement models. We therefore need to construct two different covariance estimation models that estimate process noise and three different covariance estimation models that estimate measurement noise. The five reference system Kalman filter models will in the remainder of this thesis be referred to as the road prediction, object prediction, lane marker update, vehicle update and vehicle heading update.

1.5 Structure of the report

The structure of the thesis report is as follows. Chapter 2 introduces general theory relevant to understand both the methods used and the thesis background. This includes a brief introduction to the problem of road geometry, an introduction to the Kalman filter, how the data set is created and relevant statistical theory. Chapters 3, 4 and 5 describe the different parts of the thesis work and the methods used. In these chapters, theory specific to each part or method is initially presented followed by the concepts of the methods. The covariance estimation models are evaluated and benchmarked against the baseline method based on two different performance measures on three different test sets in Chapter 6. The results presented in Chapter 6 are then discussed in Chapter 7. Lastly conclusions of the thesis work and future work suggestions are presented in Chapter 8.

2

General theory

2.1 Road geometry estimation

To provide background to the application considered in this thesis a brief introduction of road geometry estimation is given in this section.

Using the definition of road geometry given in [5] road geometry estimation may be defined as the problem of estimating the shape of the middle of a host vehicle's lane. As mentioned in Chapter 1 it is common to use Gaussian filters to perform road geometry estimation. The road geometry is thus described using a state vector $\mathbf{x}_k$ which summarizes the parameters used to define a mathematical expression of the road geometry, a mathematical road model. A Kalman filter then utilizes a process model describing the dynamics of the road geometry and a measurement model which relates measured quantities to the road geometry to estimate $\mathbf{x}_k$ at each time instance k. As seen in [6][7][8][5] the host vehicle commonly uses camera and radar sensors to measure things such as lane markers, surrounding vehicles and road-side objects which may be used to perform inference of the road geometry. A figure meant to illustrate the problem of road geometry estimation is given in Figure 2.1.

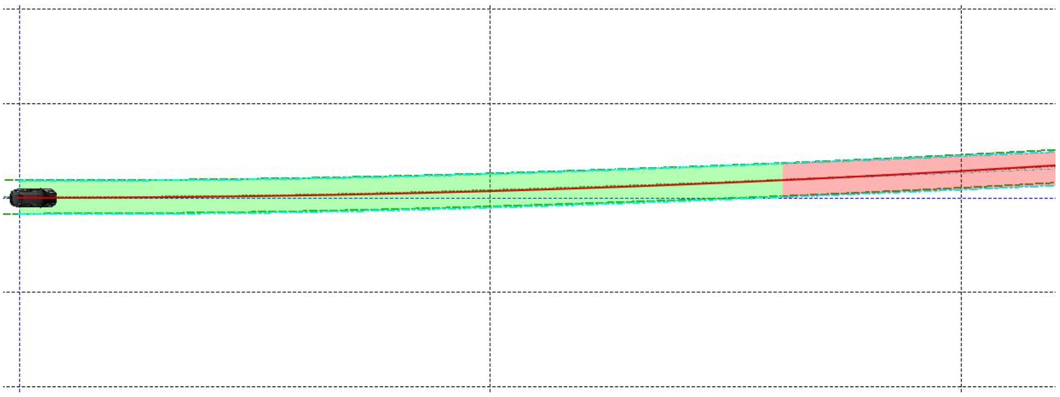


Figure 2.1: Illustration of road geometry estimation. The ground truth road is given by the dashed green lines and the road geometry estimate is described by the green and red areas.

2.2 Gaussian filters

The Gaussian distribution, also known as the normal distribution, plays a key role in Gaussian filters. The Kalman filter is one example of such a Gaussian filter. Given an initial Gaussian distribution over the state $p(x_0)$ the Kalman filter recursively computes a posterior Gaussian distribution over the state x_k at each time instance k. It does this in two steps by first calculating a predicted distribution using knowledge of the state transition dynamics and second by calculating a posterior distribution using information from an observed measurement y_k. The state is thus characterized by a state estimate $\hat{x}_k$ and estimate covariance P_k which are the mean and covariance of the Gaussian posterior distribution, respectively. The state estimate describes the estimate of the state while the estimate covariance describes the uncertainty in the state estimate.

The step in which the predicted distribution is calculated is called the prediction step which utilizes a model describing the process dynamics and the model is therefore called the process model. In the general context of Gaussian filters the process model is commonly defined as

$$\mathbf{x}_k = F_k(\mathbf{x}_{k-1}, \mathbf{u}_k) + w_k \tag{2.1}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ are the state vector and control input respectively at time k, F_k is a possibly nonlinear function describing the deterministic part of the process model at time k and w_k is a Gaussian noise process. In Kalman filters the Gaussian noise process is assumed to be a white noise process [24] which means that the samples are uncorrelated in time and zero mean, i.e., $w_k \sim \mathcal{N}(0, Q)$. As real-world dynamical systems often are complex it is not practically feasible to model the systems exactly. However, the noise process w_k captures these model errors by allowing for a distribution of possible state transitions. The process noise covariance Q thus quantifies the error of the mathematical model in (2.1) compared to the true process model. Furthermore, Q also captures the dynamics of the process model [24].

In the second step, called the update step, the posterior distribution is determined based on a model called the measurement model. The measurement model describes the relation between the state vector $\mathbf{x_k}$ and the measurements at the same time instance $\mathbf{y_k}$. In the general context of Gaussian filters the measurement model is commonly given by

$$\mathbf{y}_k = H_k(\mathbf{x}_k) + v_k \tag{2.2}$$

where H_k is a possibly nonlinear function describing the deterministic part of the measurement model at time k and $v_k \sim \mathcal{N}(0, R)$ [24]. Similarly as for w_k in the process model, v_k may model discrepancies of the mathematical model described in (2.2) compared to the true relation between the state and the measurement.

However, perhaps more importantly, real-world sensors have inherent sensor noise which also should be described by the noise term v_k.

As mentioned it is common within the literature to define the process- and measurement equations as in (2.1) and (2.2). A crucial problem with these models is that the noise process covariances, i.e., Q and R, are assumed to be fixed. In many real-world applications it is unreasonable to assume that the covariances are time-invariant. Sensor performance may vary with environmental factors and the process dynamics may change over time [24].

By allowing the noise processes in the Kalman filter model equations to be heteroscedastic, the models may more accurately describe the true dynamics. Given a vector of informative input features $\mathbf{z}_k$ that are useful in describing the heteroscedasticity of the noise processes a modified process- and measurement model may be defined as

$$\begin{aligned} \mathbf{x_k} &= F_k(\mathbf{x}_{k-1}, \mathbf{u}_k) + w_k(\mathbf{z}_k) \\ \mathbf{y}_k &= H_k(\mathbf{x}_k) + v_k(\mathbf{z}_k) \end{aligned} \tag{2.3}$$

where $w_k(\mathbf{z}_k) \sim \mathcal{N}(0, Q(\mathbf{z}_k))$ and $v_k(\mathbf{z}_k) \sim \mathcal{N}(0, R(\mathbf{z}_k))$ are input feature dependent process- and measurement noise processes, respectively. In this thesis a Kalman filter is considered and hence the process- and measurement models described in (2.3) are linear functions of the state. Nonetheless, the process- and measurement models are presented in the more general context of possibly nonlinear functions in (2.3) as to indicate that the methods considered in this thesis are not restricted to the case of linear models.

2.2.1 Toy example

A toy example is presented in this section which demonstrates the importance of correctly estimating hetereoscedastic noise in Kalman filtering.

Consider a small robot which is driving slowly along a 100m corridor. To estimate its longitudinal position, i.e., the position along the corridor, it uses a camera from which it obtains noisy observations of its absolute longitudinal position. The robot has an input signal which determines its speed and it therefore also has knowledge of the process dynamics. Once the robot passes 35m along the corridor the lights in the corridor are turned off which significantly degrades the performance of the camera sensor and its position measurements. When the robot reaches 65m the lights are turned back on and the position measurements return to being reliable. The objective of the robot is to estimate its position while driving along the full length of the corridor. Figure 2.2 demonstrates the experimental setup of the toy example.

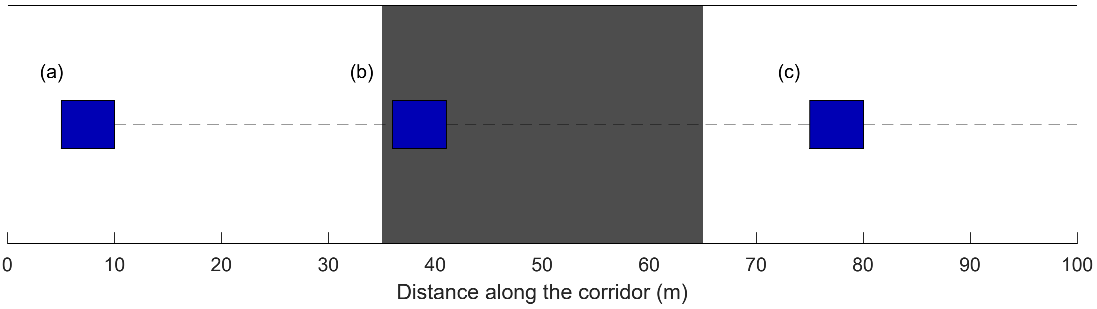


Figure 2.2: Experimental setup of the toy example seen from above. The x-axis indicates the robot's distance along the corridor, the robot is represented by the blue rectangle, the white areas correspond to bright lighting conditions and the gray area corresponds to dark lighting conditions. Point (a) the robot has started to drive along the corridor and the lights are currently turned on. Point (b) as the robot passes 65m the lights are turned off causing the position measurements from the camera to become unreliable. Point (c) the lights have been turned back on and the camera measurements may once again be trusted.

Since both knowledge of process dynamics and measurements are available the robot uses a Kalman filter to estimate its position. The process model in the Kalman filter is defined as

$$x_{k+1} = x_k + 0.1 + w \tag{2.4}$$

where x_k is the longitudinal position of the robot at time k and $w \sim \mathcal{N}(0, 0.001)$. The process model given in (2.4) is also used to generate the true state sequence and the Kalman filter process model is therefore optimal.

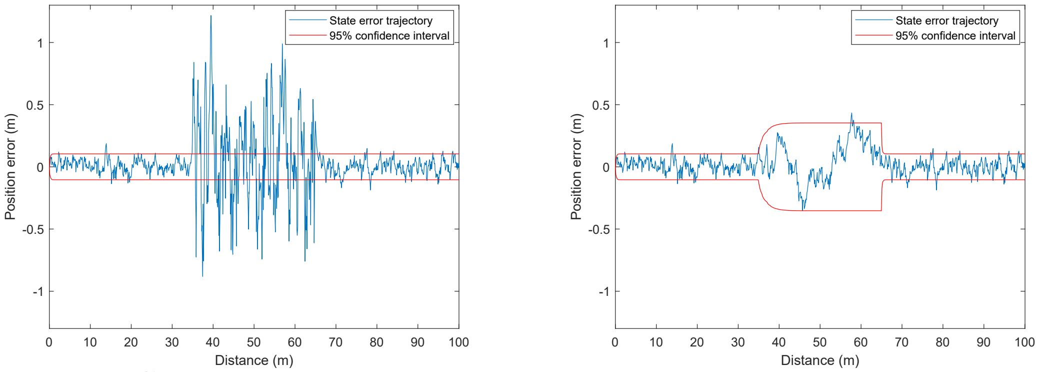
The measurement model of the Kalman filter is defined as

$$y_k = x_k + v_k \tag{2.5}$$

where y_k is the noisy measurement of the robot's absolute longitudinal position at time k and $v_k \sim \mathcal{N}(0, R_k)$. The deterministic part of the measurement model in (2.5) is optimal as it is the same as the deterministic part used to generate the measurements. However, the true measurement noise covariance is considered unknown.

Now consider two different versions of the Kalman filter; one version where the measurement noise covariance is modeled as constant and another version where the measurement noise covariance is modeled as heteroscedastic. To determine the measurement covariances for each version, a data set of ground truth measurement errors is used where the errors were predominately collected in a bright environment. In the constant version, the measurement covariance is determined by taking the sample covariance over the full data set to obtain a constant covariance R. For the heteroscedastic version the data set is divided into error samples from bright and dark environments and the sample covariance is calculated for each case separately, i.e., for the bright and dark environments, respectively. This results in a feature dependent covariance $R(z_k)$, $z_k \in \{0, 1\}$ where $z_k = 0$ indicates a dark environment and $z_k = 1$ indicates a bright environment.

The experiment described in Figure 2.2 is performed for each version of the Kalman filter and the resulting position estimation error along the corridor and the 95% confidence intervals are shown in Figure 2.3 for the constant and heteroscedastic covariances respectively.



(a) Constant covariance.

(b) Heteroscedastic covariance.

Figure 2.3: Estimated position error and 95% confidence interval for the measurement model with constant and heteroscedastic measurement covariance respectively.

From the results in Figure 2.3 it is clear that the Kalman filter which uses a heteroscedastic covariance performs better compared to the constant covariance. The heteroscedastic covariance obtains both a more consistent uncertainty with regards to the errors and also a better filter performance since the errors are smaller compared to the constant covariance. This toy example demonstrates the importance of accurately estimating heteroscedastic noise processes in Kalman filtering as it may improve both filter performance and state uncertainty estimation.

2.3 Obtaining noise samples

To determine the covariance matrix of a random vector, samples of the random vector are required. In the case of the random vectors w_k and v_k in (2.3) it is possible to obtain samples of the random vectors if one has access to ground truth state vectors. By simply solving for w_k and v_k in (2.3) one obtains

$$\begin{aligned} w_k(\mathbf{z}_k) &= \mathbf{x_k} - F_k(\mathbf{x}_{k-1}, \mathbf{u}_k) \\ v_k(\mathbf{z}_k) &= \mathbf{y}_k - H_k(\mathbf{x}_k) \end{aligned} \tag{2.6}$$

which describes realizations of the random vectors for each time instance k. A useful way to interpret the terms $F_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ and $H_k(\mathbf{x}_k)$ is to view them as predictions of the true state and a noise-free measurement respectively. Given the current ground truth state $\mathbf{x}_{k-1}$ and possible knowledge of process input $\mathbf{u}_k$ the function $F_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ strives to predict the true state vector $\mathbf{x}_k$ at the next time instant k. The term $F_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ is, thus, referred to as the predicted true state in this thesis. Similarly given the ground truth state vector $\mathbf{x}_k$ at the current time k the function $H_k(\mathbf{x}_k)$ predicts a noise-free measurement, i.e., a more accurate measurement compared to $\mathbf{y}_k$, at the same time instance k. Hence, $H_k(\mathbf{x}_k)$ is referred to as the noise-free measurement. The random vectors w_k and v_k may consequently be interpreted as the predicted state error and predicted measurement error respectively as they describe the discrepancies between the predictions and the actual values. There are several possible sources for these prediction errors. As mentioned in Section 2.2 sensors have inherent measurement noise and processes may be random in nature, e.g., a human walking around choosing directions to walk in randomly. Furthermore, the functions $F_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ and $H_k(\mathbf{x}_k)$ may not be optimal estimators in the sense that they fail to capture deterministic relations which in theory are possible to capture and in turn give rise to prediction errors, i.e., modeling errors. As there is no need to distinguish between the prediction errors in the process- and measurement models they will henceforth be referred to as simply errors. To be clear the random vector realizations w_k and v_k are henceforth referred to as errors or error samples and individual samples will be denoted as $\mathbf{e}_i$ for simplicity. Using ground truth state vector data and (2.6), one may construct a data set of errors $\mathbf{e}_i$, i.e., realizations of the random vectors w_k and v_k, and input features $\mathbf{z}_i$ as $\mathcal{D} = \{\mathbf{e}_i, \mathbf{z}_i | i = 1, 2, \ldots, N\}$.

2.4 Covariance matrix definition

As the project objective is to estimate noise coviarance matrices the definition of the covariance matrix is briefly presented for completion. A covariance matrix describes the variance of each scalar random variable and joint variability of each pairwise scalar random variables in a random vector. More mathematically given a random vector $\mathbf{X}$ the covariance matrix Σ of $\mathbf{X}$ is defined as

$$\Sigma = \mathrm{Cov}(\mathbf{X}) = E\left[(\mathbf{X} - E[\mathbf{X}])(\mathbf{X} - E[\mathbf{X}])^T\right] \tag{2.7}$$

where the operator $E()$ is the expected value and $\mathbf{X}$ is a column vector [25].

2.5 Missing data

It should be mentioned that the discussion held in this section is equally valid for both the process- and measurement models in the reference Kalman filter. Despite this the discussion is carried out by referring to the measurement model to make the discussion more concise.

As mentioned in Section 1.4 the reference Kalman filter makes use of three different update steps. In these update steps the number of measurements at each time sample may vary. This further means that the size of the covariance matrix in the measurement model may differ at each time instance. The problem of having multivariate measurements where individual elements of the measurement vector may be missing for some samples is called missing data [26]. Since the error samples in the training data sets for the measurement models are constructed using measurement samples in (2.6) the error samples consequently also suffer from missing data. This is a problem as one may not determine the covariance matrix using the sample covariance with incomplete samples. There are several possible solutions to this problem which will be discussed in the remainder of this section. These solutions are calculating pairwise covariances, discarding incomplete samples and estimating diagonal covariance matrices.

2.5.1 Pairwise covariance

One way of circumventing this problem is to calculate the pairwise covariance between each variable using the values which are not missing. This is reasonable since the definition of the covariance matrix Σ of a random variable $\mathbf{X} = [X_1, X_2, \dots, X_n]$ may be described by [27]

$$\Sigma = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \dots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \dots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & & \mathrm{Var}(X_n) \end{bmatrix} \tag{2.8}$$

where $\mathrm{Var}(X_i)$ and $\mathrm{Cov}(X_i, X_j)$ is the variance and covariance respectively of the scalar random variables X_i and X_j. However, as a result of calculating the scalar covariances pairwise for samples containing missing data the resulting covariance matrix is not guaranteed to be positive definite [28]. This is problematic as the covariance matrix needs to be positive definite to be a proper and useful covariance matrix in the Kalman filter. Consequently, the approach of calculating the covariance matrix pairwise is not suitable for the work within this thesis.

2.5.2 Discarding incomplete samples

Another option is to disregard all vector samples which contain any missing data. By removing these samples the data would no longer contain any missing data and standard techniques could be used to estimate the covariance matrix. A problem with this approach is that potentially a lot of useful data will be discarded as the valid elements in the samples with missing data also are removed. Consider the case of a measurement sample of size eight with only one missing element. Disregarding this sample would result in losing the seven valid elements which contain useful information.

A further concern with this approach is that the samples with missing data may all correspond to or correlate with a specific situation which itself may be the reason for the missing data. A Kalman filter which tracks other vehicles could for example have an update which utilizes measurements of multiple vehicles. It would then be reasonable to assume that situations with light traffic correlate with measurement samples containing only one element, i.e., a measurement of a single vehicle. Thus removing all measurement samples which contain any missing data would remove many of the samples corresponding to the situation of light traffic. This would in turn cause the model to perform poorly in cases of light traffic as this specific situation would not be represented in the training data. Hence, the approach of discarding incomplete vector samples is not used within this thesis.

2.5.3 Diagonal covariance matrix

A third alternative is to estimate a diagonal covariance matrix instead of a dense covariance matrix. As a diagonal covariance matrix only contains the variances of the individual elements in the random vector the sample elements may be used independently of each other. Consequently, the elements that are missing in each sample may be discarded without disregarding the other valid elements. After removing all missing data from the data set the covariance matrix may then simply be calculated as

$$\Sigma = \begin{bmatrix} \mathrm{Var}(X_1) & 0 & \dots & 0 \\ 0 & \mathrm{Var}(X_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & \mathrm{Var}(X_n) \end{bmatrix}. \tag{2.9}$$

It should be noted that estimating diagonal covariance matrices is not necessarily a simplification. The reason for this is that individual elements in measurement vectors in some Kalman filter updates are more appropriately modeled as independent of each other. This may be the case if the individual measurements are known to be independent in practice or they are the same type of measurement. Same type of measurement here means that multiple measurements of the same type may be obtained at each update and the order of these measurements in the measurement vector has no significant meaning. As such the individual elements are samples of the same scalar random variable and should be used collectively for estimating the variance of that random variable. In this thesis this is considered the case for the surrounding vehicles position and heading updates in the reference system Kalman filter.

Lastly as mentioned in [10] estimating diagonal covariance matrices in a model with learnable parameters may act as regularization. The reason for this is that the number of model parameters is reduced [10] since the L matrix introduced in Section 5.1.2 will simply be the identity matrix.

2.6 Regression theory and heteroscedasticity

In statistics, heteroscedasticity is present when the statistical dispersion measure (e.g. the variance) of a vector of random variables given a vector of independent variables differs across different values of the independent variable. Regression theory defines the relationship between the data and the model trained using said data. It introduces model assumptions such as zero mean valued error terms and the presence of heteroscedastic noise. In this thesis, we use common tools of regression theory, such as residual plot analysis, to perform a preliminary check on our available data and check on model assumptions such as heteroscedastic noise. However, it should be noted that these checks were not performed exhaustively for the data as there are a lot of different cases considering all five Kalman models and all the different input features used.

2.6.1 Linear regression analysis

Given a system in which variables quantities can change in random fashion, or not, it is of interest to examine the effects that some variables might be causing on others. In this context, two main types of variables can be identified: predictor variables and response variables. Predictor variables (also denominated as input variables or X-variables or regressors) are independent variables such that they can be either set to a desired value or take on an observable, but not controllable value. A change in the predictor variables has an effect on the dependent variables, i.e., the response variables (or output variables or Y-variables). In this framework, it is of interest to examine and outline the dependence that links the change in the predictor variables to the values of the response variables. In this work we will refer to $\mathbf{X}$ as the independent variable, and $\mathbf{Y}$ as the corresponding dependent variable.

A ($\mathbf{X}$,$\mathbf{Y}$) plot of the data that creates an expected value of the Y-variable for a given value of the X-variable illustratively shows the empirical relationship that links the two variables. The plot of the data pairs (X_i,Y_i) for $i = 1, 2, \ldots, n$ results in a diagram of the type shown in Figure 2.4. This type of representation is called scatter diagram.

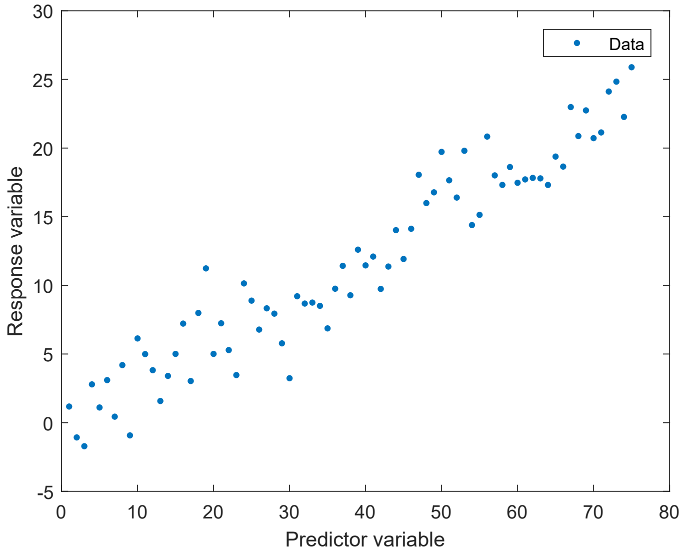


Figure 2.4: Scatter diagram of X-variable, or predictor variable, values against Y-variable, or response variable, for two random variables X, Y.

Given a data set of n measurements, a range of values can be identified for both the X-variable and the Y-variable. These values are influenced by measurement and model errors as well. A precise relationship between the two variables is not easy to identify; however, as one variable influences the other, a pattern can be outlined by considering the average observed output value for a given input value. This locus of points is known as regression curve of **Y** on **X**, i.e., $y_i = \mathrm{f}(x_i)$ for $i = 1, 2, \dots, n$; where y_i represents the i-th observation of the dependent random variable **Y** and x_i represents the i-th observation of the independent variable **X**. Similarly, a regression curve of **X** on **Y**, i.e., $x_i = \mathrm{g}(y_i)$ for $i = 1, 2, \dots, n$, can be defined. In prediction problems, this relationship can be used to estimate an average observed y_i for a given x_i; and in estimation problems it can be used to fit data to the given model in case of missing observations. The estimation of a dependent relationship between the two variables, y_i and x_i, is referred to as regression equation and can be expressed in linear form as

$$y_i = \alpha + \beta x_i + u_i \quad i = 1, 2, \dots, n \tag{2.10}$$

Parameters α and β identify a model function that links the variables. In real life applications, these parameters are unknown and can be estimated from a set of observed data $\{(x_i, y_i)$ for $i = 1, 2, \dots, n\}$ [29].

In the regression equation, u_i represents the i-th random error term. The random errors distribution is assumed to be a normal distribution with zero mean value and the errors are assumed to be independent. This linear equation represents the best-fitting straight line that better expresses the relationship between the (x_i, y_i) points for $i = 1, 2, \ldots, n$.

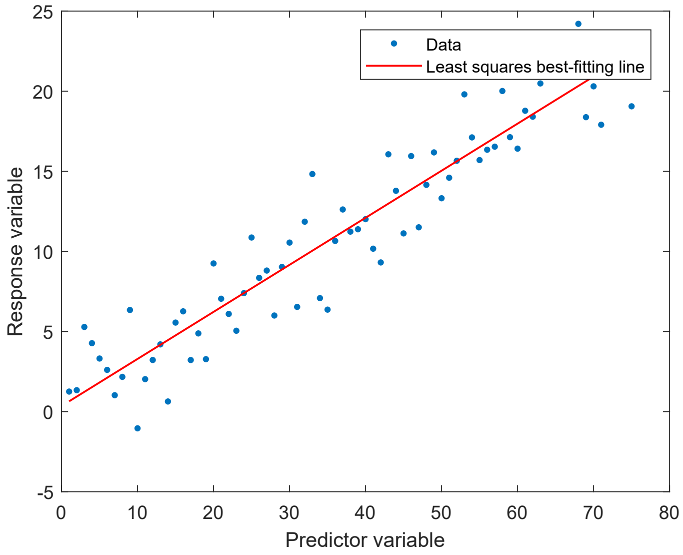


Figure 2.5: Regression curve of variables X,Y. The best-fitting straight line to model the relationship between random variables X,Y is calculated by means of least squares estimation.

The linear regression equation can be obtained by the method of least squares estimation. The result of the estimation can be seen in Figure 2.5. The residual term is defined as

$$\hat{e}_i = y_i - \hat{\alpha} - \hat{\beta}x_i \quad i = 1, 2, \ldots, n \tag{2.11}$$

The residual term $\hat{e}_i$ measures the deviation of the i-th data point (x_i,y_i) from the fitted linear regression line in the ($\mathbf{X}$,$\mathbf{Y}$) plane. The least squares estimation method evaluates the best-fitting model parameters by minimizing the sum of the residual square functions S, i.e., solving

$$\min S = \min \sum_{i=1}^{n} \hat{e}_i^2 = \min \sum_{i=1}^{n} (y_i - \hat{\alpha} - \hat{\beta}x_i)^2. \tag{2.12}$$

Solving the least squares problem for α and β gives

$$\hat{\alpha} = \overline{y} - \hat{\beta}\overline{x} \text{ and } \hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2} \tag{2.13}$$

where $\overline{y} = \sum_{i=1}^{n} \frac{y_i}{n}$, $\overline{x} = \sum_{i=1}^{n} \frac{x_i}{n}$; $y_i = \mathbf{Y_i} - \overline{y}$, $x_i = \mathbf{X_i} - \overline{x}$.

The parameters $\hat{\alpha}$ and $\hat{\beta}$ found represent the ordinary least squares estimators (OLS) of the parameters α and β, respectively. The function $\mathbf{Y} = \hat{\alpha} + \hat{\beta}\mathbf{X}$ represents the fitted model or fitted regression line. The OLS residuals $e_i = y_i$ - $\hat{\alpha}$ - $\hat{\beta}x_i$ minimize the sum of residual square functions and satisfies two important numerical properties. The first property states that (i) $\sum_{i=1}^{n} e_i = 0$, i.e., the residual sum is null; this statement implies that the ($\overline{x}$,$\overline{y}$) point belongs to the estimated regression line. The second numerical statement is (ii) $\sum_{i=1}^{n} e_i \; x_i$= 0, i.e., the residuals and the regressors are not correlated.

The least squares problem formulation relies on four assumptions [30].

Assumption 1: The first assumption is $E[u_i] = 0$ for $i = 1, 2, \ldots, n$, i.e., the random error terms have zero mean values. This assumption ensures that the average points belong to the true line.

Assumption 2: The second assumption is homoscedasticity. The variance of the disturbances is assumed to be constant, i.e., $\mathrm{Var}(u_i) = \sigma^2$, for $i = 1, 2, \ldots, n$, which implies that each observation is equally reliable.

Assumption 3: The third assumption is that the disturbances are not correlated, i.e., $E[u_i u_j] = 0$ for $i \neq j$, i,$j = 1, 2, \ldots, n$. This implies that one disturbance does not carry any information about the other disturbance terms.

Assumption 4: The fourth assumption is that the independent variable **X** is non-stochastic, and therefore not correlated with the disturbances.

2.6.2 Multiple regression analysis

Models typically include more than one regressor **X**, i.e., input variable influencing the output variable **Y**. In this case, the regression equation in linear form can be expressed as

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_K x_{Ki} + u_i \quad i = 1, 2, \ldots, n \tag{2.14}$$

where y_i is the i-th observation of the dependent random variable **Y** and x_{ki} is the i-th observation of the deterministic variable x_k for k = 1, 2, …, K.
Parameters α and β_1, β_2, …, β_K identify a model function that links the variables.

In case of multiple regression, the residual term is defined as

$$\hat{e}_i = y_i - \hat{\alpha} - \sum_{k=1}^{K} \hat{\beta}_k x_{ki} \quad i = 1, 2, \ldots, n. \tag{2.15}$$

The least squares method to estimate the linear regression equation minimizes the following residual sum of squares S

$$\min S = \min \sum_{i=1}^{n} \hat{e}_i^2 = \min \sum_{i=1}^{n} (y_i - \hat{\alpha} - \sum_{k=1}^{K} \hat{\beta}_k x_{ki})^2. \tag{2.16}$$

This results in a system of K equations in K unknowns that can be solved to find the ordinary least squares estimators (OLS), parameters $\hat{\alpha}$ and $\hat{\beta}$.

The multiple linear regression model can be written in matrix form by defining the following matrix and vectors:

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{K1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{1N} & \cdots & x_{KN} \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \alpha \\ \beta_1 \\ \vdots \\ \beta_K \end{pmatrix}, \mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix}.$$

The regression model in (2.14) can be written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}. \tag{2.17}$$

The system of equations that minimizes the residual sum of squares S in (2.16) is proved [31] to be

$$(\mathbf{X}'\mathbf{X})\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y}, \tag{2.18}$$

which leads to the least square estimators

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}. \tag{2.19}$$

The fitted values of the output dependent variable are

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \equiv \mathbf{H}\mathbf{Y}, \tag{2.20}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is known as *hat* matrix. Moreover, the residual term must satisfy

$$\hat{\mathbf{e}} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = (\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{Y} = (\mathbf{I} - \mathbf{H})\mathbf{Y}. \tag{2.21}$$

Assumptions 1–4 discussed in Section 2.6.1 are still valid in case of multiple regression analysis. Moreover, the least squares problem formulation relies on some additional assumptions.

Assumption 5: The fifth assumption is the normality assumption. The OLS estimator is also assumed to be a maximum likelihood estimator. This estimator is normally distributed and unbiased, with lower variance than any other unbiased estimator for all possible values of the parameter (minimum-variance unbiased estimator).

Assumption 6: The sixth assumption is the non-perfect multicollinearity assumption. According to this assumption, the explanatory variables are not perfectly correlated with each other, i.e., no $\mathbf{x}_k$ for $k = 1, 2, \ldots, K$ is a perfect linear combination of the other $\mathbf{x}_k$ variables. The non-perfect multicollinearity assumption is necessary to prove that there is a unique solution for the OLS estimators of the K coefficients.

2.6.3 Nonlinear regression analysis

Nonlinear regression is a particular form of regression analysis. In nonlinear regression, the response variable is modeled by a nonlinear function of the model parameters and depends on one or more independent variables, such that

$$\mathbf{Y}_i = f(\mathbf{x}_i, \boldsymbol{\beta}) + u_i \tag{2.22}$$

where f is an arbitrary nonlinear function in the components of the parameters vector $\boldsymbol{\beta}$ and $\mathbf{x}_i$ is the i-th observation on each of the independent variables. The observational data can be fitted to the model by a method of successive approximations. The random error terms u_i are assumed to be normally distributed with zero mean value and independent from one another, just as in linear regression.

The linear model (2.10) can be seen as a special case of the more general model (2.22).

Nonlinear regression is a more flexible approach than linear regression, as the function f does not need to be linear or linearizable. Nonlinear regression is a good method for cases in which linearity does not fit and the linear transformation alters model assumptions, e.g. variance homoscedasticity. The nonlinear approach provides a good tool for fitting a general and nonlinear relationship between the response variable and the predictors to the data. However, function f is required to be differentiable with respect to the elements of $\boldsymbol{\beta}$, which guarantees the existence of the least squares estimates [31]. Moreover, nonlinear regression analysis requires a precise knowledge of the relationship between the response variable and the predictors, which may be difficult or impossible to identify. The choice of an inadequate function f can result in a poor fit of the regression.

In statistics, there are several procedures for fitting nonlinear models. The nonlinear models may be: transformable nonlinear models, i.e., involving a single predictor variable $\mathbf{X}$, polynomial models, i.e., involving one or more predictor variables with terms of order higher than one, and models which are nonlinear in the parameters. The first two types of models can be fit using the linear least square method by transforming at least one of the variables (either $\mathbf{X}$, $\mathbf{Y}$ or both). The third type requires a numerical search method. In this case, a linear regression is not adequate because in these models the partial derivatives of the response variable $\mathbf{Y}$ with respect to the predictor variables involve the unknown parameters $\boldsymbol{\beta}$.

In general, there is no standard expression for the best-fitting parameters estimated, as there is in the linear case. For nonlinear models, numerical optimization algorithms can be used to determine the best-fitting parameters. A logarithmic example of nonlinear least square estimation of the best-fitting line can be found in Figure 2.6.

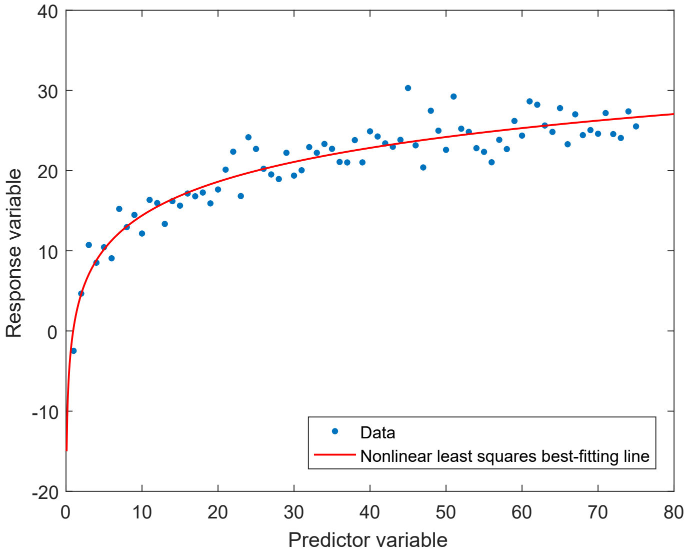


Figure 2.6: Nonlinear regression curve of variables X,Y. The logarithmic relationship between the random variables X,Y is described by $Y = a + b \log_{10} X$. The best-fitting line is calculated by means of nonlinear least squares estimation.

Under this approach, there is the assumption that the model can be approximated by a linear function, i.e., a first-order Taylor series, according to

$$f(\mathbf{x}_i, \boldsymbol{\beta}) \approx f(\mathbf{x}_i, 0) + \sum_j J_{ij}\beta_j, \tag{2.23}$$

where $J_{ij} = \frac{\partial f(\mathbf{x}_i,\boldsymbol{\beta})}{\partial \beta_j}$. Using the ordinary least square (OLS) method on the approximated model, the estimated parameters are given by

$$\hat{\boldsymbol{\beta}} \approx (\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'\mathbf{Y}. \tag{2.24}$$

The result obtained is similar to the one obtained in the linear case and described by (2.19), but using the partial derivative **J** instead of the regressor variable **X**. The linear approximation used introduces a bias.

In cases where the variance of the dependent variable **Y** is not constant over the observations, the weighted least squares approach can be used. This method, estimates the model parameters by minimizing a sum of weighted squared residuals. The weights can be iteratively estimated by the estimating algorithm, or they are usually fixed to be equal to the reciprocal of the variance of the variable **Y** at each iteration.

2.6.4 Heteroscedasticity

Heteroscedasticity, or heteroskedasticity, occurs when the variance of the error terms differs across observations. In regression theory, heteroscedasticity is the violation of the homoscedasticity assumption, according to which the variance of the disturbances is assumed to be constant. This assumption has been introduced in Section 2.6.1 as Assumption 2.

For this definition, the linear regression analysis is considered. As discussed in Section 2.6.1, the regression equation that links x_i, the i-th observation of the deterministic variable $\mathbf{X}$ and y_i , the i-th observation of the dependent variable $\mathbf{Y}$ is expressed in linear form as

$$y_i = \alpha + \beta x_i + u_i \quad i = 1, 2, \ldots, n \tag{2.25}$$

where α and β are the model parameters and u_i represents the i-th random error term.

Violation of the homoscedasticity assumption means that the random error terms have a varying variance, i.e., $E[u_i^2] = \sigma_i^2$, for i=$1, 2, \ldots, n$. This implies that each observation is not equally reliable.

Since the variance σ_i^2 depends on i, the statistical dispersion measure, i.e., the variability, of the error terms u_i are uncorrelated and the variance will vary over different groups of observations. Residual plot analysis is a useful method to detect this behaviour and the presence of heteroscedastic data.

There are several reasons why the error term in a model may have a non constant variance [32] [33]. In this thesis, we concentrate on heteroscedasticity due to the dependence of the variable we are trying to model, to the predictor variable. As the variable is random, it will have a larger variance depending on the predictor variable behaviour.

Heteroscedasticity is often the result of a non-optimal set of data. The presence of outliers, i.e., data points that diverge from an overall pattern, leads to heteroscedasticity. Including or excluding outlier samples maybe alter the results of regression analysis, especially for small data sets, where each sample has a significant weight on the overall data pattern.

2.7 Residual plot analysis

Given the fitted regression line describing the relationship between an independent variable $\mathbf{X}$ and a dependent variable $\mathbf{Y}$, defined as

$$\mathbf{Y} = \hat{\alpha} + \hat{\beta}\mathbf{X} \tag{2.26}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are the least square estimators of the model parameters; for any given value of $\mathbf{X}$, x_i, the predicted, or fitted, value of $\mathbf{Y}$, y_i is calculated as

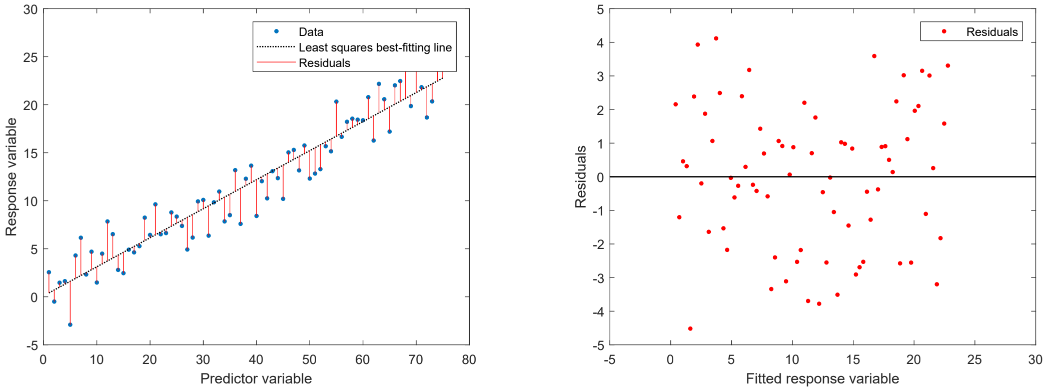
$$\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i \quad i = 1, 2, \ldots, n. \tag{2.27}$$

The deviation of the observed data point (x_i,y_i) from the corresponding predicted point on the fitted regression line $(x_i,\hat{y}_i)$ can be measured as the difference between y_i and $\hat{y}_i$. This difference is called residual and can be calculated as

$$\hat{e}_i = y_i - \hat{y}_i = y_i - (\hat{\alpha} + \hat{\beta} x_i) \quad i = 1, 2, \ldots, n. \tag{2.28}$$

The residual terms $\hat{e}_i$ can be considered as consistent estimators of the unknown error terms u_i, as they measure the same difference between the estimated and the true values of the output variable $\mathbf{Y}$ terms [34]. Moreover, the residual terms $\hat{e}_i$ are not independent.

A good tool for regression data set validation is given by the residual plot. The residual plot in the (x,y) space is a plot that has the residual terms $\hat{e}_i$ on the vertical axis versus the fitted value of the dependent variable $\hat{\mathbf{Y}}$, or the independent variable $\mathbf{X}$ on the horizontal axis, as shown in Figure 2.7.



(a) Linear regression curve.

(b) Residual plot.

Figure 2.7: Residual terms representation in a scatter diagram and in a residual plot. In the scatter diagram (a) the residual terms represent the difference between the observed values of the Y-variable and the their corresponding fitted values on the best-fit line, $\hat{Y}$. In the residual plot (b) the residuals are plotted against the fitted values $\hat{Y}$.

In particular, the residual plot is used to outline problems in the chosen data sets such as the presence of heteroscedasticity, non-linearity in the data association and presence of outliers in the data sets. In ideal conditions, a good set of data, i.e., a data set that is a good fit for regression, is so that the residual values are randomly and equally vertically spaced along the horizontal axis. The data points in the residual plot need to be randomly distributed and should not be arranged to form specific functions since the residual plot does not have a predictive value, therefore it does not enable the prediction of future data points. The opposite situation can be an indication of an unsuitable regression model.

2.7.1 Standardalized residuals

The standardized residual is a ratio used to normalize data in regression analysis.

As discussed for the general case in Section 2.6.2, the fitted values of the response variable $\mathbf{Y}$ to the regression model may be written as

$$\hat{\mathbf{Y}} = \mathbf{HY}, \tag{2.29}$$

where the fitted values have variance-covariance matrix $var(\hat{\mathbf{Y}}) = \mathbf{H}\sigma^2$ [35]. Matrix $\mathbf{H}$ is known as *hat* matrix as it maps the observed values $\mathbf{Y}$ into $\hat{\mathbf{Y}}$ ($\mathbf{Y}$-*hat*).

From the results obtained in (2.21), the residual terms can be written as

$$\hat{\mathbf{e}} = (\mathbf{I} - \mathbf{H})\mathbf{Y}. \tag{2.30}$$

Under the assumptions presented in Section 2.6, the residual terms have expected value $\mathbf{0}$ and variance-covariance matrix $var(\hat{\mathbf{e}}) = (\mathbf{I} - \mathbf{H})\sigma^2$. In details, the variance of the i-th residual term is

$$var(\hat{\mathbf{e}}_i) = (1 - h_{ii})\sigma^2, \tag{2.31}$$

where h_{ii} is the i-th element of the diagonal of matrix $\mathbf{H}$.

For models with a constant it can be demonstrated that

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_j (x_j - \overline{x})^2}, \tag{2.32}$$

and therefore it results that the value of h_{ii} is between $1/n$ and $1/r$, where n is the number of observations and r is the number of replicates of the i-th observation.

As it can be seen in (2.32), the i-th element of the *hat* matrix, h_{ii}, has a minimum of $1/n$ at the mean of $\mathbf{X}$. This implies that the variance of the fitted values $\hat{\mathbf{Y}}$ becomes smaller as the observations near the mean value. Moreover, an opposite result is observed for the residuals $\hat{\mathbf{e}}$, as the variance of the residuals is greatest near the mean value [35].

The residual plot is a valuable tool for data set validation. When analysing residual plots, it is convenient to take into account that the error variance may differ across observations. For this purpose, it is possible to refer to the *standardized* residuals $\hat{\mathbf{s}}$. Residuals can be standardized according to different statistical distributions; the most common standardization is the *internally studentized* residual, which is a form of Student's t-statistic, with the estimate of the error varying among points. In this case, the i-th elements of the standardized residual is

$$\hat{s}_i = \frac{\hat{r}_i}{\sqrt{1 - h_{ii}}\hat{\sigma}}. \tag{2.33}$$

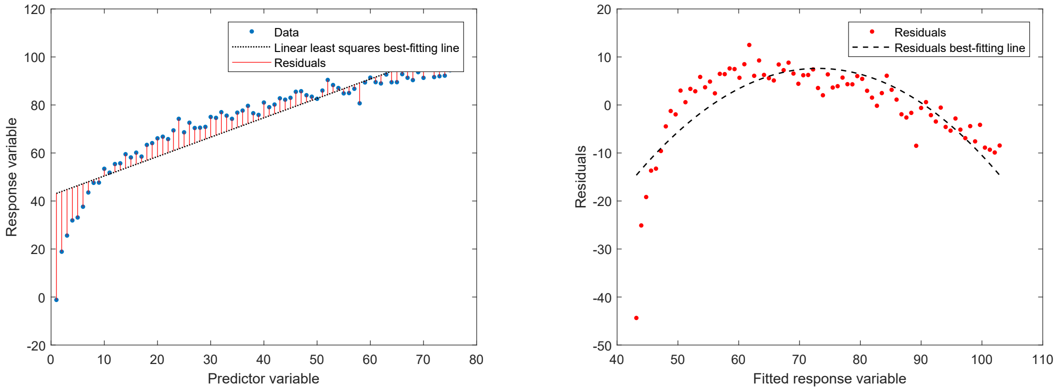
The standard deviation $\hat{\sigma}$ is estimated based on the residual sum of squares **S** described in (2.12).

In particular, standardized residual plots are used to detect the presence of outliers in the data sets. An absolute value of the standardized residual greater than two for any observation may be indication of anomalies in the data set, however this does not represent a sufficient condition to identify said observation as outlier point in the data.

2.7.2 Residual plots for nonlinear regression

Residual plot analysis is a valid tool to identify problems in the data and it is performed for nonlinear regression models in the same manner as linear regression, as the residual term represents the estimated errors in both cases.

Residual plots with parabolic shape, as all residual plots that show a specific pattern in the alignment of the data points, are a common sign of an inadequate model. Furthermore, as the model is unable to represent the relationship between the data well, the predictions will not perform well. The parabolic shape of the residual plot can be either mirrored by the regression plot, or the regression plot can identify a suitable linear regression, yet still identify an inadequate model. An example is given in Figure 2.8, where a logarithmic relationship is fitted with a linear regression line calculated by means of linear least squares. The inadequacy of the model built is mirrored in the regression plot, which takes on the typical parabolic shape for nonlinear systems fitted with linear models.



(a) Linear regression curve.

(b) Residual plot.

Figure 2.8: The scatter diagram (a) shows the linear regression curve for the nonlinear system (X,Y). In the residual plot (b) the residuals are plotted against the fitted values $\hat{Y}$ and they identify a typical parabolic shape.

This particular type of residual plot leads to three possible conclusions. First, plots that show such patterns may indicate that a variable needs to be transformed. Second, the pattern may be caused by a missing variable. Third, if the pattern has a clear parabolic shape, nonlinear regression is the best choice for model regression. Therefore, residual plots can be used to identify when the linear regression is not sufficient to produce a good model and nonlinear regression would be more appropriate.

In particular, a good strategy of nonlinear regression is to add a squared term of the regressor $\mathbf{X}^2$ in the model, for a better chance to fit the parabolic curve. This type of approach can be extended to other shapes, e.g. the S-shaped curve, by adding a cubic term of the regressor $\mathbf{X}^3$, although it is a less common approach.

2.7.3 Residual plots for categorical data

Categorical variables are all variables which take a finite number of values, or categories. More details on this type of variables are given in Chapter 3.

We consider now input variables $\mathbf{X}$ with $k > 2$ where k is the number of categories. Categorical terms cannot be used directly in regression. This type of variable can be included in a regression model by creating $k-1$ dummy variables $\mathbf{X}_i$, $i = 1, 2, \dots, k-1$. These new variables are numerical variables and can therefore be used in regression, and they take value 1 for all units that belong to the category of interest, and 0 otherwise, such that

$$\mathbf{X}_i = \begin{cases} 1 & \text{for category } i, \\ 0 & \text{otherwise.} \end{cases} \tag{2.34}$$

The last category, for which a dummy variable is not created, is the reference category and the parameters of all dummy variables are interpreted with respect to this category. This transformation from categorical to numerical data is known as *one-hot encoding* and will be described more in Chapter 3. Input variables $\mathbf{X}$ with $k = 2$, can be considered as already hot-encoded, as they already present only two possible categories, therefore values.

The relationship that links the dependent variable $\mathbf{Y}$ and the dummy variables $\mathbf{X}_i$ can be shown in a plot on the $(\mathbf{X},\mathbf{Y})$ axis as per all regression relationships discussed in the prior sections. To have a better understanding of the data, the data points may be divided in subgroups according to the values taken by the reference category. A fitted model can be defined for each subgroup of data.

The residual terms that derive from the regression analysis can be plotted against the finite values of the categorical independent variable. This results in subgroup of points in vertical lines for each category. Ideally, all lines are centered in zero and follow a bell-shaped distribution with similar standard deviations.

3

Feature Selection

As mentioned in Section 1.1, features relevant for estimating process- and measurement noise, such as features describing the road type being driven or environmental conditions e.g. rain, may be interesting to use in the models. At the start of this thesis a large set of candidate input features were available. However, not all features are useful for estimating the process and measurement noise and the number of input features used can not be too large since the models are used in a real-time application.

Feature selection, or variable subset selection, is the technique of selecting a subset of relevant features out of a larger features set, to use in model construction. The purpose of feature selection is to remove from the original set of data irrelevant features that do not contribute to the prediction variable, in order to build a more accurate model. The optimal subset of features is characterized by the least number of dimensions that contribute the most to prediction accuracy [36]. Feature selection plays a key role in machine learning as well as in several prediction methodologies, as training a model on irrelevant data can negatively impact model performance in a significant way. In this work, feature selection is introduced in order to select a subset of relevant features that prove to be useful to build a good covariance estimator.

There are many benefits to feature selection techniques: simplify the data for better visualization and interpretation, reduce the training utilization times, reduce the requirements in terms of needed measurements and storage, avoid the curse of dimensionality [37]. Feature selection also aims at reducing the problem of overfitting. An overfitted model is a model that contains a higher number of parameters than it is expected by the given data. Overfitting implies that noise has been interpreted as underlying part of the model structure; this often happens in the presence of redundant features. However, subset of useful features may also include redundant features, i.e., features that add no relevant information to other features [37]. In spite of the fact that feature selection is prone to eliminating redundant features to guarantee dimensionality reduction, some redundant features may be kept since they carry important information when combined with a correlated relevant feature.

3.1 Theory

3.1.1 Filter methods for feature selection

Feature selection methods are classified in three main categories based on the way the method combines the feature selection algorithm and the model building: filter, wrapper and embedded methods. In this work, the filter approach was used to perform feature selection.

Filter methods for feature selection select the variables regardless of the model built [38]. The variables are selected based on general characteristics of the training data such as statistical dependence or distance between classes [39]. The algorithm discards the least interesting variables from the original set of candidate input features. The advantages of filter methods are that they are not expensive in terms of required computation time and they are robust to overfitting [40]. Moreover, they reach a better generalization as the results are independent from any predictor [38]. Filter methods can be limited in cases for which the method does not take in consideration the relationship between variables [40].

Filter methods tend to select a high number of features, and therefore a threshold is required in order to pick the right subset [40]. Different approaches to evaluate the best relevant features lead to several indices for ranking and feature selection. Filter methods perform feature selection based on statistical tests of four types: consistency metrics, distance metrics, mutual information, and correlation. In this thesis, correlation coefficients are used as a measure of the dependency of the features to the error variances. The most common techniques for correlation measure are based on correlation coefficients such as Pearson's correlation coefficient and Spearman's rank coefficient. Both these methods are discussed in Sections 3.1.2 and 3.1.3 respectively.

3.1.2 Pearson correlation coefficient

The Pearson correlation coefficient is a measure of linear correlation between two random variables. Denoting the random variables **X** and **Y** it may be defined as

$$\rho_{\mathbf{X},\mathbf{Y}} = \frac{\operatorname{cov}(\mathbf{X},\mathbf{Y})}{\sigma_{\mathbf{X}}\sigma_{\mathbf{Y}}} = \frac{E\left[(\mathbf{X}-\mu_{\mathbf{X}})(\mathbf{Y}-\mu_{\mathbf{Y}})\right]}{\sigma_{\mathbf{X}}\sigma_{\mathbf{Y}}} \tag{3.1}$$

where $\mu_{\mathbf{X}}$ and $\mu_{\mathbf{Y}}$ are the means and $\sigma_{\mathbf{X}}$ and $\sigma_{\mathbf{Y}}$ are the standard deviations of **X** and **Y** respectively.

3.1.3 Spearman's rank correlation coefficient

Spearman's rank correlation coefficient is a measure of monotonic correlation between two random variables. Denoting the random variables **X** and **Y**, it can be defined as the Pearson correlation coefficient between the corresponding rank variables $rg_{\mathbf{X}}$ and $rg_{\mathbf{Y}}$

$$r_s = \rho_{rg_{\mathbf{X}},rg_{\mathbf{Y}}} = \frac{\operatorname{cov}(rg_{\mathbf{X}}, rg_{\mathbf{Y}})}{\sigma_{rg_{\mathbf{X}}}\sigma_{rg_{\mathbf{Y}}}} \tag{3.2}$$

where ρ is the Pearson correlation coefficient applied to the rank variables, $\operatorname{cov}(rg_{\mathbf{X}}, rg_{\mathbf{Y}})$ is the covariance of the rank variables and $\sigma_{rg_{\mathbf{X}}}$ and $\sigma_{rg_{\mathbf{Y}}}$ are the standard deviations of the rank variables $rg_{\mathbf{X}}$ and $rg_{\mathbf{Y}}$ respectively.

The Spearman's coefficient is a nonparametric measure of the statistical dependence, i.e., the correlation, between the rankings of two variables.

3.1.4 Moving variance correlation

Ideally one would like to compute the correlation between the individual continuous input features z and the corresponding error variances σ^2 to perform feature selection. This would require a data set of pairs of input features and error variances $\mathcal{D}_\sigma = \{\sigma_i^2, z_i | i = 1, 2, \dots, N\}$ which in reality one does not have access to. Instead what is available in this thesis is a data set of pairs of input features and error samples, i.e., $\mathcal{D} = \{e_i, z_i | i = 1, 2, \dots, N\}$ which will be discussed further in Section 3.2. One may think of this data set as a space of features, a feature space, where each feature value z_i has a location within the space and a corresponding error value e_i.

To perform feature selection using the data set $\mathcal{D}$ we propose to use the sample variance to calculate local variances $\hat{\sigma}^2$ for each z_i in the feature space. This is done by computing the sample variance for errors corresponding to features in a neighbourhood of each z_i. One may then obtain a data set of local variances and features $\mathcal{D}_{\hat{\sigma}} = \{\hat{\sigma}_i^2, z_i | i = 1, 2, \dots, N\}$. The data set $\mathcal{D}_{\hat{\sigma}}$ can then be used to calculate correlation measures to perform feature selection.

Furthermore, we propose to calculate multiple data sets $\mathcal{D}_{\hat{\sigma},j}$ of varying neighbourhood sizes. The reason for this is that there is no obvious choice of any single neighbourhood size since we do not know how quickly the true variance σ^2 changes with the feature value. A correlation measure may now be calculated for each data set $\mathcal{D}_{\hat{\sigma},j}$ e.g. the correlation measures presented in Sections 3.1.2 and 3.1.3. If any of these correlation measures are larger than a threshold τ then the feature may be interesting to include in the covariance estimation model. The value of the threshold τ is problem dependent as one may expect varying correlation strengths depending on the problem. What is important is to find the features which correlate the most with the variance among the features one considers.

There is a straightforward way to implement the feature selection method described in this section. Since both the error and the feature are scalars one may order the error samples in order of increasing value of their corresponding feature. A moving variance may then be applied to the ordered sequence to obtain the local variances.

A moving variance is an algorithm which calculates variances over a sliding window. Calculating local variances for different neighbourhood sizes is then simply done by applying moving variances of different window sizes. The algorithm is summarized as pseudo-code in Algorithm 1.

Algorithm 1: Moving variance correlation

```
Result: isUsefulFeature
isUsefulFeature = false;
set $\tau$;
set window sizes;
order errors and features based on increasing feature values;
for window sizes do
    $\hat{\sigma}_j^2$ = apply moving variance to error sequence;
    $\rho_j$ = calculate correlation between $\hat{\sigma}_j^2$ and features;
    if $\rho_j \geq \tau$ then
        isUsefulFeature = true;
    end
end
```

It should be noted that the neighbourhood size may not be chosen arbitrarily large since if all other feature samples are contained in the neighbourhood around each z_i then all the estimated variances $\hat{\sigma}_i^2$ will be the same.

3.1.5 One-hot encoding for categorical data

Many machine learning algorithms, as well as regression theory, require all input and output variables to be numerical. This limitation does not effect the algorithm itself, but it is a constraint to its implementation. As a consequence, all categorical variables must be converted to a numerical form. The concept of categorical data is further discussed in Section 3.3.1.

One-hot encoding is a technique of data representation which ensures all entries of a vector or bits of a string take the value of 0, except for a single one, that takes value 1. In statistic, this technique is used for representing categorical data.

The categorical variable can be replaced by a numerical variable which takes value 1 to indicate whether a certain category of the original variable is present for that observation, and 0 otherwise. Given i possible categories, these replacement variables are known as dummy variables, and they can be expressed as

$$\mathbf{X}_i = \begin{cases} 1 & \text{for category } i, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

as presented in (2.34). An advantage of the one-hot encoding technique is that it does not mirror any type of ordinal relationship between categories, as opposite to encoding techniques such as integer encoding, which assign an integer value to each category. Integer values have a natural ordered relationship between each other which is learned by any algorithm and leads to inaccurate results.

Let's consider a simple example. A categorical variable indicating weather the weather is sunny or not, can assume two values, labels "yes, the weather is sunny" and "no, the weather is not sunny". This categorical variable generates two binary variables, called *dummy variables*, corresponding to each of the two categories. Considering the first dummy variables, for each observation, this numerical variable will take value 1 if the weather is sunny, and 0 otherwise. The same can be done for the second dummy variable. In particular, each observation takes value 1 for only one of the two dummy variables at a time, as the two conditions are mutually exclusive, i.e., the weather is either sunny, or not. Therefore, to encode a categorical variable with two labels, only one of the binary variables is needed. The choice of which of the two replacement variables to use is not important, as they both contain the same information and are sufficient to represent the original categorical variable.

This concept can be extended for categorical variables with k labels. Let us reprise the previous example, and consider a new, more complex categorical variable representing the current weather conditions. This variable contains k labels, e.g. "Sunny", "Rainy", "Clouded". According to the previous discussion, only $k - 1$ dummy variables are needed to represent the information of the original categorical variable. Furthermore, each observation takes value 1 for only one of the dummy variables and 0 for all others, according to the one-hot encoding, i.e., if the weather is sunny, it cannot be rainy, or clouded at the same time. The one-hot encoding technique with $k-1$ dummy variables allows to represent the original information with one less dimension. In practice, if an observation takes value 0 for all the dummy variables, then it will take value 1 for the k-th omitted category.

In linear regression and for all machine learning algorithms, including neural networks, categorical variables are transformed in numerical variables by means of the one-hot encoding technique. This is possible as linear regression and machine learning methods have access to all features during training, and it allows to keep the correct number of degrees of freedom, $k - 1$, yet represent the whole original categorical information.

3.2 Data set

As described in Section 2.3 it is possible to construct a data set of errors and input features $\mathcal{D} = \{\mathbf{e}_i, \mathbf{z}_i | i = 1, 2, \dots, N\}$ using ground truth state vector data. As the reference system considered in this thesis has five different Kalman filter models we construct five different data sets, one for each Kalman filter model. There is, however, no reason to distinguish between these five data sets as the methods used in this thesis are applied equivalently to all five data sets and independently of the other data sets. For conciseness we therefore refer to a single data set during many parts of the report. But the ideas and methods discussed are equally applicable to all five data sets.

The ground truth state vector data used in this thesis is derived from data sampled from multiple driving sessions for a total amount of ground truth data corresponding to approximately 14.75 hours of driving. A large majority of the driving sessions are from highway scenarios and highways are consequently the focus of this thesis. The data set was divided into three parts; a training set, a development set and a test set. Approximately 2.25 hours of the data was assigned for the test set. The driving sessions in the test set are presented more in detail later in Section 6.2. The remaining 12.5 hours of data was divided into the training and development sets. The training set was used for training the models and consisted of approximately 95% of the remaining data. The development set consisted of approximately 5% of the remaining data and was used for evaluating different model choices and tuning of the models. It is crucial to have a development set as it enables the model designer to assess how well the model generalizes to unseen training data during model construction.

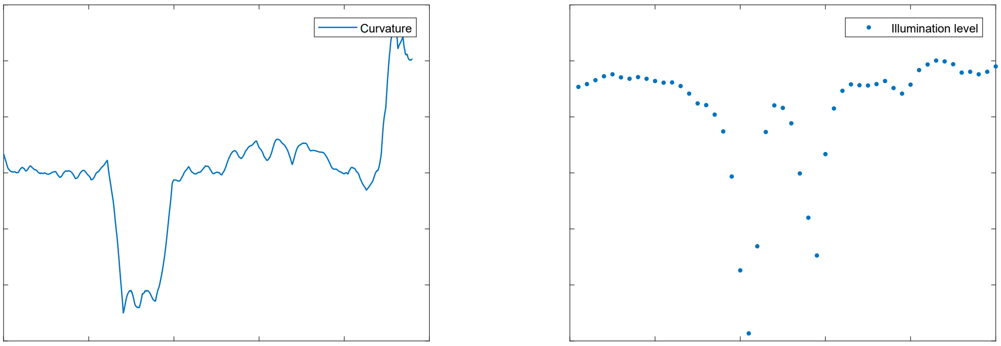
3.3 Model assumptions check

This section is dedicated to testing for data anomalies in the data set provided and the fulfillment of our modelling assumptions. The analysis is not carried out extensively over all Kalman filter models, but represents instead a first check on the quality of our data and noise model assumptions such as heteroscedasticity. Moreover, some conclusions are drawn on using residual plot analysis as a feature selection tool. A general regression model describes the relationship between the output variable Y and the independent input variables X. In this thesis, the error terms, i.e., the difference between the real output variable and its observed value for each time instant, is assumed to be normally distributed, with zero mean and varying variance. The error terms $\mathbf{u}$ for our system are unknown, nevertheless the errors can be estimated by means of the residuals $\hat{\mathbf{e}} = \mathbf{Y} - \hat{\beta}\mathbf{X}$, i.e., the difference between the observed value of Y and its estimated value. The two terms are interchanged in this Section to describe the same concept of distance of the observed output from its true or estimated value. Further details on the definition of residual are discussed in Section 2.7. In this thesis we discuss the benefits of modelling non-constant noise covariances in the Kalman filter model equations to describe the dynamics of the system under study more accurately. This approach comes from the observation that the error terms are indeed heteroscedastic, as it is proven in the analysis carried out in the following sections. Furthermore, input features which lead to a strong heteroscedastic behaviour in the error terms, are considered to produce a better estimation performance when included as inputs to the covariance estimation models.

3.3.1 Input features overview

The uncertainty estimation in road geometry at the basis of this thesis depends on a multitude of different factors. Measurement signals describing these factors are called features. The sensor performance, and therefore the filter performance, may be effected by external factors such as bad weather conditions and light conditions and the road geometry dynamics may depend on the type of road. These features are inputs of the models built in this thesis. The data set used in this work consists of data sampled from multiple driving sessions.

The information contained in the input features to our models can assume different forms, as the data is sampled and stored in different types of variables. The most crucial distinction to be made is between numerical variables and categorical variables. Numerical variables have either a meaning as a measurement, or they keep count of a quantity, thus they are also known as quantitative variables. Numerical data can be further distinguished into two types: discrete data and continuous data. Discrete data is used to represent a finite number of items that can be counted. Therefore, the possible values that these variables can assume can be either finite or countably infinite, i.e., go from 0, 1, 2 up to infinity. Continuous data is used to represent measurements. The possible values that these variables can assume cannot be counted. A convenient way to describe these variables is by means of intervals of value on the real numbers $\mathbb{R}$ line.



(a) Continuous feature. **(b)** Discrete feature.

Figure 3.1: Examples of a continuous variable (a) and a discrete variable (b).

In statistics, categorical variables are variables that can take on a limited, usually fixed, number of possible values, and assign each observation to a specific group, or nominal, category on the basis of a qualitative property. Categorical variables can take two or more values. Categorical variables that take on exactly two values are called binary or dichotomous variables. Categorical variables that can take more than two values are called polytomous variables. Polytomous variables can be further categorized in ordinal and nominal variables, based on whether the categories can or cannot be ordered or ranked. Categorical variables can take on numerical values, each corresponding to a different category, or non-numerical values, usually terms which are significant to the meaning of each category. The numerical values of categorical data do not have any mathematical meaning, thus categorical data is also known as qualitative data.

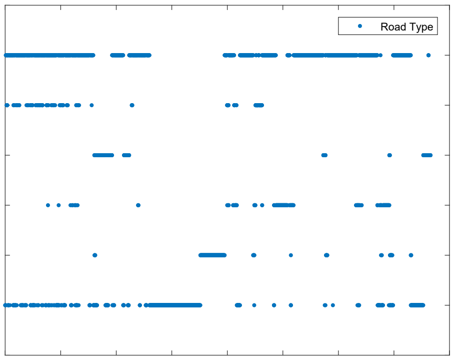
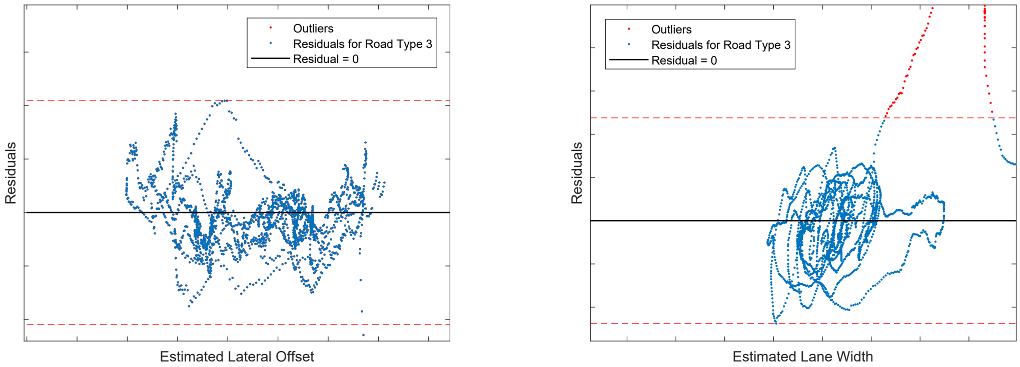


Figure 3.2: Example of categorical variable.

Histogram, scatter plots and most graphical ways to plot statistical data require that the data is numerical for the plot to make sense. One way to overcome this problem is to assign an identificative characteristic to the data sample that belong to the same category, e.g. color, representation style. However, this solution is restrictive and works well only for simple data representation. To be able to use statistical tools, categorical data needs to be transformed into numerical data. An example of transformation is the one-hot encoding described in Section 3.1.5.

3.3.2 Data quality check

Residual plot analysis is a useful tool to detect anomalies among the data, such as the presence of outliers. An outlier is a data point that diverges from the overall data pattern in a specific sample. The outlier points can influence the relationship between the dependent and independent variables and therefore, the presence of outliers in the training data, may generate a faulty model. There are several reasons behind the presence of outliers in the data, and most of them are case-specific, i.e., due to some unexpected event that happened while collecting the data. Outliers in the X-direction are considered influential observations. Influential observations derive from observations that are unusually large or diverge extremely from the center of the reference data distribution. Outliers can be easily recognized in a scatter plot of the independent variable against the dependent variable or in a plot of the residuals against the fitted values of the dependent variable, as the behaviour is often mirrored from the data to the residuals. An example can be seen in Figure 3.3b. Moreover, outlier observations are characterized by a standardized residual value larger than 3. The definition of standardized residual is introduced in Section 2.7.1.



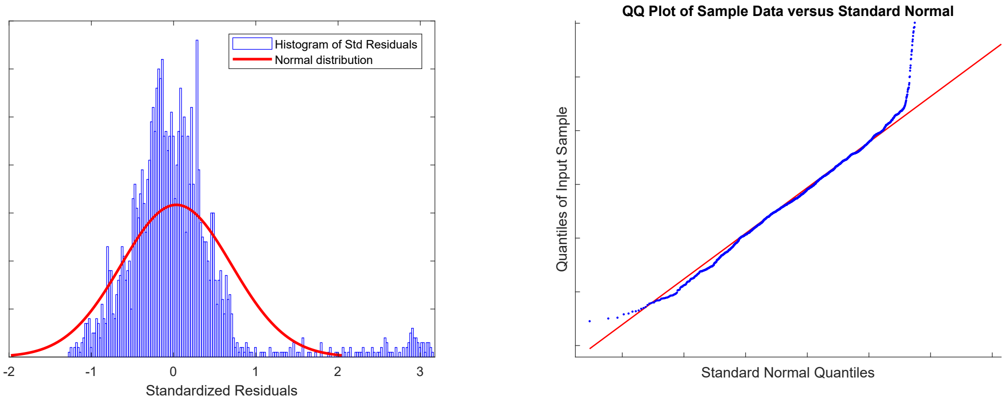
(a) No outliers.

(b) Outliers.

Figure 3.3: Residual plots to show the absence and presence of outliers respectively. The residual plot against the estimated Lateral Offset for Road Type 3 shows no outliers outside the confidence interval. The residual plot against the estimates Lane Width for Road Type 3 shows several outlier residual terms.

3.3.3 Normality assumption

The model error is assumed to follow a normal distribution. This assumption can be verified by means of residual plot analysis [34], where the residual term is the estimated value of the error term. In order to compare the estimated residuals to the scale of a standard normal distribution, we refer to the standardized residuals. The error distribution can be outlined from the plot of the histogram of the standardized residuals. To check the normality assumption, a QQ-plot can be used, i.e., a scatter plot which displays two sets of quantities against one another. In the QQ-plot, a theoretical standard normal distribution is plotted against the standardized residuals. If the residuals are normally distributed, the points in the QQ-plot lie along a straight diagonal line, the bisecting line.



(a) Histograms of standardized residuals.

(b) QQ-plot of standardized residuals.

Figure 3.4: Normality check for standardized residual of the Lane Width when Road Type is 3. The normality condition worsens for higher values of the standardized residual.

For the example shown in Figure 3.4, the normality assumption is fulfilled, yet the condition is not strictly followed and the quality worsens for higher values of the standardized residuals. Statistical measures such as confidence intervals and tests of hypothesis rely on the normality assumption. However, small deviations from this assumption do not represent an issue in terms of estimation.

3.3.4 Zero mean assumption

The zero mean assumption states that the random error terms have zero mean values. This result means that the average deviation of each error term from the true model is zero. Checking on the fulfillment of the zero mean assumption can be easily achieved by means of a residual plot of the residuals $\hat{\mathbf{e}}$ against the fitted output variable $\hat{\mathbf{Y}}$ or the regressors $\mathbf{X}$. If the assumption is true, the data plot is approximately symmetric around zero, as shown in Figure 3.5. This assumption implicitly assumes that the error terms are uncorrelated and that the average points of the output variable belong to the fitted line, i.e., $E(\mathbf{Y}) = \beta\mathbf{X}$. If the assumption is not fulfilled, the residual plot will show a systematic, asymmetrical pattern that deviates from the zero line.

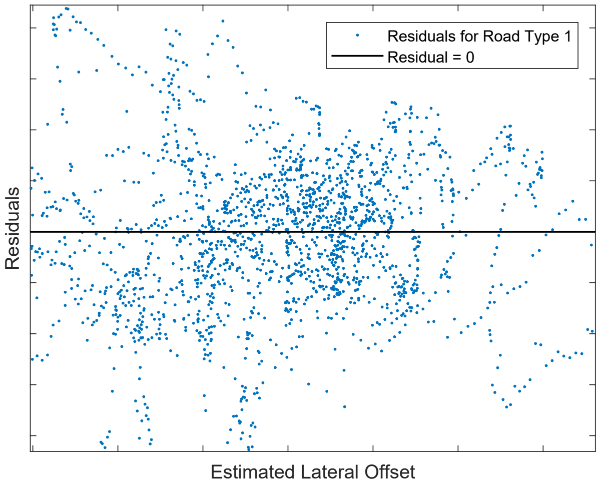
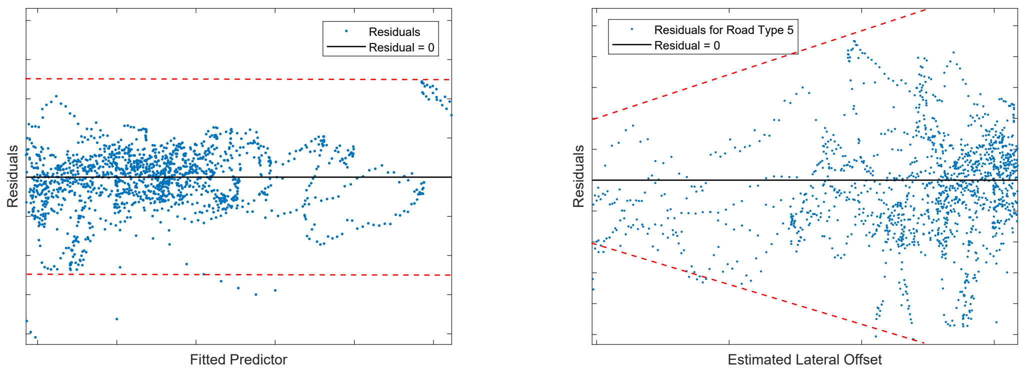


Figure 3.5: Scatter plot of Lane Width for Road Type = 5. The plots is symmetric with respect to the 0 value of the residuals.

The zero mean assumption, along with the assumption that the regressors $\mathbf{X}$ are independent from the error terms, implies zero conditional mean. In statistics, it can be proven that given the independence between $\mathbf{X}$ and $\mathbf{u}$, the mean value of $\mathbf{u}$ given $\mathbf{X}$ is equal to the mean value of $\mathbf{u}$, i.e., $E[\mathbf{u}|\mathbf{X}] = E[\mathbf{u}]$. Given that the zero mean assumption is respected, i.e., $E[\mathbf{u}] = 0$, the presence of both assumptions implies zero conditional mean, $E[\mathbf{u}|\mathbf{X}] = 0$. While respecting the two assumptions presented above is a sufficient condition for zero conditional mean, the opposite is not true. The correlation between the independent variable and the error, also known as *endogeneity*, implies that the independent variable can be used to predict the error term. In real-time applications, this is often the case. As discussed for what concerns the models introduced in this thesis, this type of information can be included in the estimation model and the error can be modeled as heteroscedastic, which is more conforming to real data observations and makes the noise and regressors uncorrelation unnecessary. There are different reasons why the strict independence of the regressors from the errors may fail, and consequently the zero conditional mean is not achieved. In regression theory, this may happen because some variables have been omitted during the model building, which correlate with one of the regressors. In this case, an omitted-variable bias (OVB) is present, as an important casual factor has been left out. As a consequence, errors may occur in the estimation and the effect of the included variables may be over- or underestimated. Checking on the zero conditional mean condition is not as straightforward as for other regression assumptions. The nonzero mean of the errors may be absorbed by the model, resulting in zero mean residuals. As a consequence, a check can not be carried out on the common mean of the residuals. A check can be performed on the plot of the residuals against the predictors. Even if the residuals may have a zero mean on average, conditionally they may have means some distance from zero. The single residual means do not form a smooth curve, yet the curve still tends to sit above the zero line.

3.3.5 Heteroscedasticity check

The heteroscedasticity check is carried out to prove that the homoscedasticity assumption is not fulfilled, i.e., the variance of the error terms is not constant. If the variance σ^2 depends on i, the error term u_i will assume different values for different groups of observations. In this check, as for the check on the normality assumptions, the standardized residuals are employed to assess properties of the errors. To detect the phenomenon of heteroscedasticity, the fitted values of the dependent output variable $\hat{\mathbf{Y}}$ are plotted against the standardized residuals. If the scatter plot is random, i.e., reproduces scattered data randomly distributed along the 0 axis, the homoscedasticity assumption is fulfilled. Otherwise, if there is a pattern in the plot, the errors are heteroscedastic, as a systematic trend indicates heteroscedasticity. A systematic trend in the plot displays higher or lower variability for higher or lower values of $\hat{\mathbf{Y}}$. This funnel-shaped pattern in the residuals plot is typical of heteroscedasticity, as shown in Figure 3.6b.



(a) Homoscedasticity.

(b) Heteroscedasticity.

Figure 3.6: Heteroscedastic noise is detected by the presence of a funnel-shaped trend of residuals distribution symmetric to the 0 horizontal axis.

The statistical consequences of the presence of heteroscedastic noise are, as for the other assumptions aforementioned, the possibility of incorrect results in confidence intervals and tests. From an estimation point of view, and in the Kalman filter application in particular, the presence of heteroscedastic noise is not a major problem, if the error is modelled accurately. For this purpose, modelling the noise covariance as heteroscedastic is proved to lead to a more consistent uncertainty with regards to the error as well as a better filter performance. This concept finds major relevance in the work carried out in this thesis.

Several statistical tests can be carried out to detect heteroscedasticity, e.g. the

White Test. Most of these tests evaluate criteria based on the knowledge that the ordinary least squares (OLS) estimator of the model parameters β is consistent even in the presence of heteroscedastic errors. Therefore, the OLS residuals will mirror the heteroscedasticity of the true disturbances. The tests can be applied to these residuals and still detect the phenomenon. Residual-based tests are robust as they are able to detect a variety of forms of heteroscedasticity. However, these tests may be less effective depending on the estimation model adopted. As the task of describing the system through a perfect model is highly difficult to achieve to perfection, using model-based tests can lead to incorrect conclusions on the shape of the errors. In this context, a graphical residual-based approach for checking on the homoscedasticity assumption tends to be more convenient and effective.

3.3.6 Residual analysis for feature selection

Data analysis is one of the preliminary phases of any type of model building for estimation. Data analysis interprets and presents the data into useful information that provide context for the data itself. In this thesis work we were faced with the challenge of selecting input features for noise modelling out of a much larger set. In this context, residual plot analysis, as well as (X,Y) plots, represent a simple and efficient tool for data screening and to visualize statistical properties for linear and nonlinear estimation models. The input features selected come from the idea that uncertainty estimation in road geometry depends on a multitude of external factors, such as weather conditions and road type. These features, which enter our model as inputs, have an effect on the uncertainty as they affect the sensors performance and road dynamics. The features selected from this intuition can be easily tested through residual plot analysis and regression analysis to have an indicative measure on how these observations weight on the uncertainty estimation. Furthermore, plotting the data enables to recognize unexpected trends that may be present.

Residual plot analysis is useful to identify problems in the data, such as the presence of outliers that might impact the training data set and lead to wrong estimators. Furthermore, this type of plots are used to identify data sets that are not good candidates for regression. In this thesis work, we exploit the idea of modelling the measurement- and process covariances as heteroscedastic, on the basis that the filter noise terms are heteroscedastic as well in real life applications. Regression- and residual plot analysis are a straightforward way to prove this idea. Moreover, input features which exhibit a strong heteroscedastic behaviour may be exploited as inputs to the covariance estimation models.

3.4 Input features selected

At the start of this thesis, a large number of possible input features was available. As introduced in Section 2.1, the host vehicle uses camera and radar sensors to take measurements regarding factors such as lane markers, surrounding vehicles, road-side objects, which are used by the Kalman filter to obtain the road geometry estimate. The estimation uncertainty that we propose to model as heteroscedastic noise in this thesis, is therefore influenced by the sensors performance. Moreover, sensor performance may be affected by environmental factors and the road geometry dynamics may depend on the characteristics of the road being driven. Features describing these factors may be informative for estimating the noise covariances in the filter. The feature selection techniques introduced in this chapter were used to select a suitable number of relevant input features for noise model estimation. Some examples of input features selected are the distance to lane markers, the host car velocity and the source of the measurement, i.e., if the measurement was obtained from a camera, radar or if it is a fused measurement derived from both a camera and radar measurement. It is important to note that the number of input features used in the noise estimation methods described in the following chapters can be increased by introducing new relevant features, according to the estimation requirements and limitations of the methods.

4

Discrete Covariance Estimation

The principal idea of the discrete covariance estimation method is to divide the training data into different cases generated from all the possible combinations of the corresponding input feature values. The covariance for each case may then be calculated using the sample covariance. This results in a discrete model which maps from cases in a discrete space to covariance matrices. The domain of the model function may thus be described as a finite set of integers $\mathbf{z} \in \mathbb{Z}^n$ where n is the number of input features. The set is finite in the sense that each element of $\mathbf{z}$ may only have a finite set of values depending on the number of cases for that specific input feature. In this thesis this model is referred to as Discrete Covariance Estimation (DCE).

4.1 Theory

4.1.1 Sample covariance

The sample covariance is the estimator of the population covariance. The population is the data set from which a sample of data on one or more random variables is taken. The sample covariance matrix is a square matrix: its i, j-th element is the sample covariance between the sets of observed values of two of the variables; the i, i-th element is the sample variance of the observed values of one of the variables. For a single observed variable, the sample covariance is a 1×1 matrix, i.e., a single number, containing the sample variance of the observed values of said variable.

The sample covariance matrix $\mathbf{S}$ is a $K \times K$ square matrix, where K is the number of observations for each observations data vector. Given the observations vector x_i, the entries of the matrix are defined as

$$s_{jk} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k), \tag{4.1}$$

where the entry s_{jk} is the estimate of the covariance between the j-th and k-th variables of the data population. The matrix can be expressed in vector form as

$$\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T. \tag{4.2}$$

Column vector $\overline{\mathbf{x}}$ is the sample mean vector whose j-th element $\overline{x}_j$ represents the average value of the N observations of the j-th variable x_{ij}

$$\overline{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} \quad i = 1, 2, \dots, N. \tag{4.3}$$

The sample covariance definition was defined in [41] in matrix form as

$$\mathbf{S} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}_N^T)(\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}_N^T)^T, \tag{4.4}$$

where $\mathbf{X}$ is the matrix of the N observations vectors, $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N]$. Vector $\mathbf{1}_N$ is by definition the $N \times 1$ vector of all ones.

The sample covariance matrix is a positive semi-definite matrix. Note that for any matrix A, the A^TA matrix is positive semi-definite. Therefore this property can be easily checked by rearranging the sample covariance matrix accordingly: if the observation vectors are arranged as rows instead of columns,

$$\mathbf{S} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{M} - \mathbf{1}_N\overline{\mathbf{x}}^T)^T(\mathbf{M} - \mathbf{1}_N\overline{\mathbf{x}}^T), \tag{4.5}$$

where $\mathbf{M}$ is the $N \times K$ matrix whose column j is the vector of N observations on variable j. Moreover, if the rank of the $(\mathbf{x}_i - \overline{\mathbf{x}})$ vectors is K, the matrix is positive definite.

4.1.2 Bessel's correction for unbiasedness

Given a random row vector $\mathbf{x}_i = [x_{i1} \ \ x_{i2} \ \dots \ x_{iK}]$ where the j-th element x_{ij} for $j = 1, 2, \dots, K$ is one of the random variables, the sample covariance is an unbiased estimate of the covariance matrix of said vector $\mathbf{x}_i$ [41]. The condition of unbiasedness is given by Bessel's correction. This statistical correction consists of using $N-1$ instead of N in the sample variance and covariance formulas, with N being the number of observations. As described in (4.4), the sample covariance is calculated from the difference between each observation and the sample mean, i.e., the estimator of the population mean. However, the sample mean is defined based on all observations and therefore it is correlated to each of them to some extent. Given the actual population mean $E(\mathbf{X})$, the unbiased entry of the sample covariance matrix is

$$s_{jk} = \frac{1}{N}\sum_{i=1}^{N}(x_{ij} - E(\mathbf{X}_j))(x_{ik} - E(\mathbf{X}_k)), \tag{4.6}$$

where the coefficient in the denominator is N, the number of observations. It is important to state that, both $\frac{1}{N}$ and $\frac{1}{N-1}$ tend to $\frac{1}{N}$ for large values of N, thus the standard sample covariance is approximately equal to the unbiased sample covariance estimate when the population sample is large.

4.2 Model

The discrete covariance model introduced in this Chapter may be described as an n-dimensional discrete space, where n is the number of features. The set is a finite space of integers, as each element $\mathbf{z} \in \mathbb{Z}^n$ may only take a finite number of values. All the possible combinations of fixed values taken on the input features at the same time, define a finite set of possible cases. The DCE model then maps each case to a specific measurement- or process noise covariance. The intuition behind this space definition is that, the covariance matrices can take on different values according to a multitude of different external, measured factors which, combined, identify our input features cases.

As discussed in Section 2.1, the host vehicle uses camera and radar sensors measurements to perform inference of the road geometry. Such measurements commonly include lane markers, target vehicles or road-side objects. Moreover, several sensors in the host vehicle are used to measure external factors they may affect the estimation uncertainty in the road geometry estimation. These measurements are the input features to our covariance estimation model, as introduced in Section 1.1. From intuition, input features measuring factors such as the weather conditions, the lighting conditions affect the sensors performance, as the sensors performance worsens in hostile driving conditions such as bad weather. Moreover information such as the type of road the vehicle is driving on may help understand the correspondance between the characteristics of the road and a certain road geometry dynamic.

The input features considered in this model may describe information in different forms and they may enter the DCE model as either numerical or categorical variables.

4.2.1 Continuous and discrete features

As mentioned in Section 2.7 the input features may be of varying types. As the model constructs the noise covariances matrices from a discrete number of cases, where each case depends on the values of the input features in the input set, said features are required to take on a finite number of possible values. Discrete features are easily handled as the values are distinctly separated. The discrete inputs may therefore be divided into cases by simply considering each value of the discrete variable as a unique case. To comply with the discrete domain of the model function continuous features also need to be divided into cases, i.e., the continuous inputs need to be discretized. For continuous inputs, however, the division of the features into different cases is not as clear. The reason for this is that continuous features are real valued and form a continuum in which values are not distinctly separated. A sensible way to account for a continuous feature is to map the values from the original large continuous set to a smaller set with a finite number of elements. For this purpose, a simple equal-width discretization approach can been applied to separate all possible continuous values into n number of intervals of the same width. The difference between the values included in this new small set and the original input feature values represents a loss of information. However, the input feature is still sufficiently described for the purposes of this study. It is interesting to mention that categorical variables can be introduced in the DCE model as input features as discrete numerical variables by means of transformation rules such as the one-hot encoding discussed in 3.1.5.

4.2.2 Discrete cases definition

The model maps a finite number of discrete values into a space set of possible cases, defined by the values taken by the variables in the input features set. To cover each location in this space, the data is divided into a number of cases N_c to be covered equal to

$$N_c = \prod_{i=1}^{n} c_i, \tag{4.7}$$

where c_i is the number of unique values that the i-th input feature can take and n is the number of input features. The model thus estimates a covariance matrix for each possible scenario. Each input feature may describe different scenarios for each of its values, e.g. the covariance matrix assumes different values for the road type being a country road or a highway. Let's now consider the example of two discrete input features that take c_i and c_j possible discrete values in the considered system. As each of them defines c_i and c_j scenarios respectively, the total number of possible scenarios, therefore covariance model cases, is given by $c_i c_j$. Hence, for n different features, the total number of scenarios is given by the product in (4.7).

A limitation of this method therefore is that it becomes intractable for a large number of input features since each case will be covered by very little data in the data set which makes the sample covariance inaccurate. Moreover we can observe how the number of cases scales up as either the number of features values or the number of features increases.

In the first case, the number of cases N_c increases as a product with the number of possible values for each input feature. This outcome may not be critical for simple implementations. However, dividing the features into a larger number of cases may be an interesting approach to describe the cases space more accurately, but results in even more divisions of the data, thus less sample observations for each unique discrete value and an unbalanced training data set. This limitation has to be taken in consideration especially in the case of continuous features, where the number of discrete values in which to divide each continuous feature is chosen arbitrarily.

In the second case, N_c scales up quickly as the number of features increases, as it does so exponentially. Let us consider a set of n input features with two possible cases each. The number of total combinations is given by $N_c = 2^n$. This result can be critical as it means the model requires a lot of data to be able to cover each different possible case since each case becomes very specific, i.e., a combination of specific values for each input feature describes a very specific scenario, which may happen rarely in real life. Let us consider a simple set of three input features describing the weather conditions, the type of the road and the level of illumination. The case described may be very specific even with this limited number of input as it may be difficult to find enough data in the training data set for the specific situation, e.g. raining, nighttime and driving on an urban road, all at the same time. This limitation doesn't subsist if the training data available is exhaustive. However, for the training data set used in this thesis, this is not the case.

As a result of this discussion, it can be acknowledged that the discrete covariance estimation method does not generalize to unseen data in view of the fact that the absence of data from a specific case implies that a covariance estimate for said case cannot be calculated. DCE has low computational complexity and memory requirements with respect to the parametric methods discussed later in this thesis, PCE and DeepCE. However, the parametric methods that will be introduces in Chapter 5 are able to generalize to cases not seen in the training data, which the discrete method can not.

4.2.3 Model implementation

The covariance estimate for each input case is the unbiased sample covariance described in (4.2) and repeated here for convenience

$$\mathbf{S} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{e}_i - \bar{\mathbf{e}})(\mathbf{e}_i - \bar{\mathbf{e}})^T \tag{4.8}$$

where the vector $\mathbf{e}_i$ is an error sample from the data set described in Section 3.2.

As introduced in Section 1.4 the Kalman filter reference system used in this thesis implements one prediction step for road and object prediction, and three update steps. Therefore we need to construct two covariance estimation models to estimate the process noise covariances and three covariance estimation models to estimate the measurement noise covariances.

In some cases it might be interesting to estimate dense covariance matrices as there may exist correlations between different measurements or between different state vector elements. An example of this is in the lane marker update where the position of lane markers at different distances are measured. It may then be reasonable to assume that the measurement noise of measurements corresponding to lane markers close to each other are correlated. However, in the different update steps the number of measurements at each time sample may vary. Moreover, in the object prediction step, the state vector may vary in size at each time step. As a consequence, sample covariance cannot be calculated to obtain a dense covariance matrix. To overcome this problem, three main solutions are proposed in Section 2.5. In the DCE method, a diagonal covariance matrix is estimated instead of a dense covariance matrix. The advantage is that the diagonal of the covariance matrix contains the variances of the individual elements of the random vector, thus missing data elements may be discarded without disregarding other valid elements, as they are independent from one another.

Because of the problem discussed in Section 4.2.2, of quickly increasing number of discrete cases with increasing number of possible values of features and number of features, the continuous input features that were used were not discretized. Instead, the discrete covariance estimation model described in this chapter models heteroscedasticity from the continuous input features by means of noise model matrices. These noise model matrices contain the values of the continuous input features and are multiplied with the covariance matrix to produce a final covariance matrix estimate. The model equations are thus given by

$$\begin{aligned} \mathbf{x_k} &= F_k(\mathbf{x}_{k-1}, \mathbf{u}_k) + N_{w,k} w_k(\mathbf{z}_k) \\ \mathbf{y}_k &= H_k(\mathbf{x}_k) + N_{v,k} v_k(\mathbf{z}_k) \end{aligned} \tag{4.9}$$

where N_w and N_v are the noise model matrices and $w_k(\mathbf{z}_k) \sim \mathcal{N}(0, Q(\mathbf{z}_k))$ and $v_k(\mathbf{z}_k) \sim \mathcal{N}(0, R(\mathbf{z}_k))$. The DCE model thus estimates the initial covariances $Q(\mathbf{z}_k)$ and $R(\mathbf{z}_k)$ and the final covariance estimates are constructed using the noise model matrices. From the property $\mathrm{Cov}(A\mathbf{X}) = A\mathrm{Cov}(\mathbf{X})A^T$ it follows that the final covariance estimates, corresponding to the process- and measurement models described in (2.3), are given by $N_{w,k}Q(\mathbf{z}_k)N_{w,k}^T$ and $N_{v,k}R(\mathbf{z}_k)N_{v,k}^T$, respectively. Consequently, the continuous features also have an effect on the covariance matrix.

In other words, in the Kalman filter, the process- and measurement noise terms w_k and v_k are multiplied with noise model matrices. This changes the process- and measurement models slightly as shown in (4.9) which also means that the random variable realization calculations described in (2.6) need to be modified by multiplying with the inverse of the noise model matrix. The initial covariance matrix estimates may then be determined using the sample covariance as described before. This solution has relevant effects on the final estimation results and it allows to handle continuous features more efficiently. In Section 4.2.2 the consequences of dividing the input features into several different cases has been discussed. This consideration has particular relevance for continuous features. One of the limitations of the discrete method proposed is that it does not allow to have a large number of input features, as that would mean generating cases which are too specific, thus leading to some cases where we have little to no data at all. In this proposed solution, the continuous features are accounted for by the noise model matrices while the discrete features are accounted for in the discrete covariance estimation model proposed in this chapter.

5

Parametric models

There seem to be varying definitions of what constitutes a parametric model in the literature. In this thesis a parametric model is defined as a model which can be described by a fixed number of parameters. Parametric approaches are of interest in this work as they provide an efficient way of estimating covariances since the information contained in the training data may be summarized by the parameters. Additionally, the number of parameters is constant once the model has been constructed which means that the evaluation time of the model is constant. This is an important property within road geometry estimation as the covariance estimation is performed online with real-time requirements.

Based on the definition given in the previous paragraph two different parametric models are presented in this section as they share similarities and important underlying concepts. In this thesis these models are referred to as Parametric Covariance Estimation (PCE) and Deep Covariance Estimation (DeepCE). In the following sections theory relevant for both methods is initially presented followed by a description of each of the models. It should also be noted that the estimated covariance is frequently referred to as R in this section to simplify notation. The concepts are, however, equally applicable to the estimated process noise Q in the process models.

5.1 Approach

5.1.1 Objective function

One of the fundamental problems of modeling a feature dependent covariance of a random variable $\mathbf{X}$ from samples of $\mathbf{X}$ is that one does not have access to the true covariances. This differs the problem of learning a covariance model compared to the common problem in supervised learning where one has access to ground truth labels of the variable of interest. The model may then be optimized using some distance measure between the estimated value and ground truth. Given the true distribution of the random variable $\mathbf{X}$ it would, for example, be possible to minimize the Kullback-Leibler divergence between the true and the estimated distribution as mentioned in [23].

An alternative approach is instead to maximize the likelihood of observing the errors, i.e., the errors in the data set $\mathcal{D}$ described in Section 3.2, from the estimated distributions [10]. Assuming that the errors are Gaussian distributed the distribution of the errors is given by the multivariate normal distribution

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{m}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{e}-\mu)^T\Sigma^{-1}(\mathbf{e}-\mu)} \tag{5.1}$$

which describes the probability of observing the sample $\mathbf{e}$ given a mean and covariance, i.e., $p(\mathbf{e}|\mu, \Sigma)$. Assigning each individual error sample $\mathbf{e}_i$ a specific covariance matrix R_i and assuming that the errors are zero mean the likelihood objective may be defined as

$$\arg\max_{\phi} \mathcal{L}(R_i|\mathbf{e}_i) = \arg\max_{\phi} \mathcal{L}(f(\mathbf{z}_i)|\mathbf{e}_i), \quad i = 1, 2, \ldots, N \tag{5.2}$$

where f is the function mapping from features $\mathbf{z}_i$ to covariance matrices R_i, i.e., $R_i = f(\mathbf{z}_i)$, and ϕ is the set of learnable parameters of the function f. The function f will also be used to impose the constraint that the estimated covariance matrices are positive definite which will be discussed in detail in the next section, Section 5.1.2. Furthermore, f will act as a regularizer for the estimation of the covariance matrix which will be discussed later in Section 5.1.3. The learnable parameters ϕ are later defined specifically for PCE and DeepCE, in Sections 5.2.2 and 5.3.2, respectively. As the natural logarithm is a monotonically increasing function the objective in (5.2) may be reformulated as

$$\arg\max_{\phi} \log\left(\mathcal{L}(f(\mathbf{z}_i)|\mathbf{e}_i)\right) = \arg\max_{\phi} \ell(f(\mathbf{z}_i)|\mathbf{e}_i), \quad i = 1, 2, \ldots, N \tag{5.3}$$

where ℓ denotes the log-likelihood. By further adding a minus sign the objective may be described as a minimization problem

$$\arg\min_{\phi} -\ell(f(\mathbf{z}_i)|\mathbf{e}_i), \quad i = 1, 2, \ldots, N \tag{5.4}$$

where $-\ell$ is the negative log-likelihood. Assuming that the errors are independent the likelihood of observing a collection of errors $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N\}$ is given by the product of their individual probabilities

$$\mathcal{L}(\{R_1, R_2, \ldots R_N\}|\{\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_N\}) = \prod_{i=1}^{N} p(\mathbf{e}_i|R_i). \tag{5.5}$$

Substituting (5.1) and (5.5) into (5.4) and utilizing the assumption that the means are zero one obtains

$$
\begin{aligned}
&\arg\min_{\phi} -\log\left(\prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{m}{2}}|R_i|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{e}_i^T R_i^{-1}\mathbf{e}_i}\right) \\
&= \arg\min_{\phi} \sum_{i=1}^{N} \left(\frac{m}{2}\log(2\pi) + \frac{1}{2}\log|R_i| + \frac{1}{2}\mathbf{e}_i^T R_i^{-1}\mathbf{e}_i\right)
\end{aligned}
\tag{5.6}
$$

where the objective may be simplified by disregarding the constant term and the scaling resulting in

$$
\begin{aligned}
&\arg\min_{\phi} \sum_{i=1}^{N} \left(\log|R_i| + \mathbf{e}_i^T R_i^{-1}\mathbf{e}_i\right) \\
&= \arg\min_{\phi} \sum_{i=1}^{N} \left(\log|f(\mathbf{z}_i)| + \mathbf{e}_i^T f(\mathbf{z}_i)^{-1}\mathbf{e}_i\right).
\end{aligned}
\tag{5.7}
$$

Lastly the optimization is subject to the constraint that the estimated covariances matrices R_i are positive definite as to produce proper and useful covariances. The constrained optimization problem is thus described by

$$
\begin{aligned}
&\arg\min_{\phi} \sum_{i=1}^{N} \left(\log|f(\mathbf{z}_i)| + \mathbf{e}_i^T f(\mathbf{z}_i)^{-1}\mathbf{e}_i\right) \\
&s.t. \quad f(\mathbf{z}_i) \succ 0.
\end{aligned}
\tag{5.8}
$$

5.1.2 LDL decomposition

The constraint of positive definiteness in (5.8) may be difficult to handle as it is not obvious how one may enforce this constraint in the optimization. However, as proposed by the authors in [10] the constraint may be relaxed using the LDL decomposition resulting in constraints which are more easily handled.

The LDL decomposition decomposes a square positive definite matrix R into a lower unit triangular matrix L and a diagonal matrix D as

$$
R = LDL^T = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \ddots & 0 \\ L_{ij} & \dots & 1 \end{bmatrix} \begin{bmatrix} \ddots & 0 & 0 \\ 0 & D_{ii} & 0 \\ 0 & 0 & \ddots \end{bmatrix} \begin{bmatrix} 1 & \dots & L_{ji} \\ 0 & \ddots & \vdots \\ 0 & 0 & 1 \end{bmatrix}.
\tag{5.9}
$$

If the scalar elements in D are constrained to be positive then the LDL decomposition exists and is unique for all positive definite matrices [23]. In this thesis the nonzero elements in L and D are denoted as $\boldsymbol{l}$ and $\boldsymbol{d}$ respectively. To relax the constraint in (5.8) the function f is redefined to map features $\mathbf{z}$ to a vector of the elements $\boldsymbol{l}$ and $\boldsymbol{d}$ instead of a covariance matrix. Consequently, the only constraint on the function f is that the scalar elements $\boldsymbol{d}$ should be positive. This constraint may be imposed as in [10] by applying an element-wise exponential function to the elements $\boldsymbol{d}$ meaning that the diagonal elements of D is now similarly as in [23] given by $\exp(\boldsymbol{d})$ instead of $\boldsymbol{d}$ where exp() is an element-wise exponential function. The covariance matrix may then be constructed from the elements $\boldsymbol{l}$ and $\boldsymbol{d}$ using (5.9).

Using the LDL decomposition as in [10], the constrained optimization problem in (5.8) may then be relaxed resulting in the unconstrained problem

$$\arg\min_{\phi} \sum_{i=1}^{N} \Big(\log|L_i D_i L_i^T| + \mathbf{e}_i^T (L_i D_i L_i^T)^{-1} \mathbf{e}_i \Big) \tag{5.10}$$

where the parameters ϕ are the parameters of the function f which maps features $\mathbf{z}$ to $\boldsymbol{l}$ and $\boldsymbol{d}$.

As explained in [23] one does not necessarily have to use the LDL decomposition specifically as other matrix decompositions which solve the problem of positive definite matrices are also applicable. However, as argued by the authors in [23] the LDL decomposition further has the benefit of numeric stability in calculating the log-determinant in (5.10). This can be seen by considering [23]

$$\log|LDL^T| = \log|L| + \log|D| + \log|L^T| = \log|D| \tag{5.11}$$

where the $\log|L|$ terms are zero as the determinant of a unit triangular matrix is one. Remember that the elements of the diagonal matrix D is given by $\exp(\boldsymbol{d})$ which means that (5.11) may be further simplified as [23]

$$\log|D| = \log\left(\prod_{i=1}^{N} \exp(\boldsymbol{d}_i) \right) = \sum_{i=1}^{N} \log\left(\exp(\boldsymbol{d}_i)\right) = \sum_{i=1}^{N} \boldsymbol{d}_i \tag{5.12}$$

where the sum is taken over all elements of $\boldsymbol{d}$. The sum in (5.12) thus provides a numerically stable way of calculating the log-determinant.

5.1.3 Regularization

Regularization is essential when using the optimization objective described in (5.10). This follows from the fact that (5.10) is derived from the likelihood given by (5.5) and that the errors are assumed to be zero mean. Namely, it can be shown that if each error corresponds to a unique point in the feature space then the optimal covariance matrices will be underconfident. This statement will be proved by a derivation in this section and lastly a regularization technique will be suggested which mitigates the problem of the optimization objective.

The derivation may be started by considering that if each error $\mathbf{e}_i$ corresponds to a unique point in the feature space $\mathbf{z}_i$ then each error sample may have a specific covariance matrix $R_i = L_i D_i L_i^T$. The optimization problem in (5.10) may then be written as

$$\arg\min_{\phi} \sum_{i=1}^{N} \left(\log|R_i| + \mathbf{e}_i^T R_i^{-1} \mathbf{e}_i \right). \tag{5.13}$$

As the covariance matrices R_i may be chosen independently for each error sample $\mathbf{e}_i$ the sum in (5.13) is minimized by minimizing each term in the summation separately. These independent terms are given by

$$\log|R_i| + \mathbf{e}_i^T R_i^{-1} \mathbf{e}_i. \tag{5.14}$$

To find the minimum of (5.14) with respect to the covariance R_i one needs to find the stationary point of the expression. To find this stationary point one may set the derivative with respect to R_i^{-1} equal to zero and solve for R_i [42]. The variable substitution $M_i = R_i^{-1}$ is thus made for clearer notation. The expression in (5.14) may now be rewritten as

$$\log|M_i^{-1}| + \operatorname{tr}\left(\mathbf{e}_i^T M_i \mathbf{e}_i \right) \tag{5.15}$$

where tr() is the matrix trace and the equality $\operatorname{tr}\left(\mathbf{e}_i^T R_i^{-1} \mathbf{e}_i \right) = \mathbf{e}_i^T R_i^{-1} \mathbf{e}_i$ holds since $\mathbf{e}_i^T R_i^{-1} \mathbf{e}_i$ is a scalar [42]. Furthermore, because of the cyclic property of the matrix trace [42] the expression may be reformulated as

$$\log|M_i^{-1}| + \operatorname{tr}\left(\mathbf{e}_i \mathbf{e}_i^T M_i \right). \tag{5.16}$$

The derivative of (5.16) with respect to M_i may then be derived by utilizing the following properties

$$\begin{aligned} \frac{\partial}{\partial A} \operatorname{tr}(BA) &= B^T \\ \frac{\partial}{\partial A} \log|A| &= A^{-T} \\ \log|A^{-1}| &= -\log|A| \end{aligned} \tag{5.17}$$

where the first two properties are derived in [42] and the last expression follows from properties of the determinant and logarithm.

The derivative of (5.16) is then given by

$$\begin{aligned}
&\frac{\partial}{\partial M_i}\left(\log|M_i^{-1}| + \operatorname{tr}\left(\mathbf{e}_i\mathbf{e}_i^T M_i\right)\right) \\
&= \frac{\partial}{\partial M_i}\left(-\log|M_i|\right) + \frac{\partial}{\partial M_i}\operatorname{tr}\left(\mathbf{e}_i\mathbf{e}_i^T M_i\right) \\
&= -M_i^{-T} + (\mathbf{e}_i\mathbf{e}_i^T)^T \\
&= -M_i^{-1} + \mathbf{e}_i\mathbf{e}_i^T
\end{aligned} \tag{5.18}$$

where $M_i^{-T} = M_i^{-1}$ since M_i is symmetric. Setting the derivative in (5.18) equal to zero and remembering that $M_i = R_i^{-1}$ results in

$$-R_i + \mathbf{e}_i\mathbf{e}_i^T = 0. \tag{5.19}$$

Solving for R_i in (5.19) gives the covariance which minimizes (5.14), i.e., the maximum likelihood covariance for the unique error sample $\mathbf{e}_i$,

$$R_i = \mathbf{e}_i\mathbf{e}_i^T. \tag{5.20}$$

The optimal covariance matrix for each unique error sample $\mathbf{e}_i$ is thus given by $R_i = \mathbf{e}_i\mathbf{e}_i^T$ which is the outer product of the error with itself. It is easy to see that this results in an underconfident covariance in the scalar case. Consider a scalar error e_i. The optimal covariance R_i of e_i is given by

$$R_i = e_i^2 \tag{5.21}$$

which has the standard deviation $r_i = \sqrt{R_i} = e_i$. Since the errors are assumed to be zero mean Gaussian distributed approximately 68% of the errors should be within one standard deviation of zero. However, as shown in this section if each error corresponds to a unique point in the feature space then the optimal covariances will be given by $R_i = e_i^2$ in the scalar case. These covariances are underconfident as all errors e_i in the training set are precisely within one standard deviation $r_i = e_i$ of the mean instead of only approximately 68%. It has hence been shown that if each error corresponds to a unique point in the feature space then the optimal covariances will be underconfident.

It should be noted that the above derivation is based on the presumption that each error sample corresponds to a unique point in the feature space. However, for real data, i.e., data obtained from the real world, this is a reasonable assumption as real measurements always contain noise. Depending on the precision of the measurements some of the measurements may have the same value but a majority of the measurements will most likely have unique values.

One way to view the fact that the optimal covariances in the Gaussian likelihood objective are underconfident is that the training data is noisy. In other words the training labels in the data set are not perfect since they in combination with the objective do not describe the true covariances exactly. As to not overfit the model to the noisy data it is important to use regularization. One option is to use L2 regularization which penalizes larger parameter values in the model. Using L2 regularization and by denoting R_i as $R_i = R_i(\phi)$, to make it clearer that R_i is a function of ϕ, the optimization objective in (5.13) may be reformulated as

$$\arg\min_{\phi} \sum_{i=1}^{N} \Big(\log|R_i(\phi)| + \mathbf{e}_i^T R_i(\phi)^{-1} \mathbf{e}_i \Big) + \lambda \|\phi\|_2^2. \tag{5.22}$$

5.2 Parametric Covariance Estimation

A parametric covariance estimation method was presented in [10] which maps from features to covariance matrices by a parametric model. There are both strengths and weaknesses in using parametric models. If the number of parameters in the model is constant then evaluating the model for covariance prediction is also constant in time. This is crucial in real-time systems such as the reference Kalman filter in this thesis as predictions can not be too slow. However, as described in [23] a weakness of the parametric model in [10] is that it assumes a specific parametric form of the function mapping features to covariances.

5.2.1 Positive definiteness constraint

As explained in [10] and discussed in Section 5.1.1 and 5.1.2 the parametric model has to predict positive definite matrices as to produce useful and proper covariances. The auhtors of [10] propose to use the LDL decomposition in combination with the exponential function as discussed in Section 5.1.2 to handle this positive definiteness constraint. As such the estimated covariance R is given as in (5.9) here repeated for convenience

$$R = LDL^T. \tag{5.23}$$

Furthermore, the exponential function may cause problems during optimization as large optimization steps may cause the exponential elements to overflow. We solve this in the same way as in [10] by normalizing the features to an interval of -1 to 1.

5.2.2 Model form

A crucial part of a parametric model is the parametric form used to express the model. The parametric form used in [10] to produce the elements of the D and L matrices is an exponentiated weighted sum and weighted sum of the input features respectively described as

$$
\begin{aligned}
D_{ii}(\mathbf{z};\phi) &= \exp(w_i^T\mathbf{z}) \\
L_{ij}(\mathbf{z};\phi) &= v_{ij}^T\mathbf{z}
\end{aligned}
\tag{5.24}
$$

where D_{ii} corresponds to the ith diagonal elment of the D matrix, L_{ij} corresponds to the element on the ith row and jth column in the L matrix, w_i and v_{ij} are the corresponding weight vectors constructed from the model parameters ϕ and $\mathbf{z}$ is the vector of input features. Relating (5.24) to the notation used in Section 5.1.2 the elements of $\boldsymbol{d}$ and $\boldsymbol{l}$ are given by the weighted sums $w_i^T\mathbf{z}$ and $v_{ij}^T\mathbf{z}$ respectively.

A possible criticism of the parametric model in (5.24) is that the weighted sum does not explicitly make use of a constant bias term. However, the authors of [10] make use of a constant feature in the feature vector which allows the weighted sum to apply a bias. Consequently, the criticism is not valid.

It is also worth mentioning that by using a parametric model the model inputs are not restricted to have discrete values as in the Discrete covariance estimation method. The domain of the model function is instead given by the set of real numbers $\mathbf{z} \in \mathbb{R}^n$ where n is the number of input features.

5.2.3 Objective function

The authors of [10] model the heteroscedastic noise process as a zero mean Gaussian with an input dependent covariance. Using a data set of errors the model is then optimized based on the Gaussian log-likelihood of the error samples given the estimated covariances. In this work the negative log-likelihood is used instead of the log-likelihood as we prefer to minimize a loss function. However, these objective functions have the same aim, i.e., to maximize the likelihood of the error samples given the estimated covariances.

5.2.4 Objective regularization

As discussed in Section 5.1.3 regularization is important when using the log-likelihood objective for covariance estimation. In [10] L2 regularization is used in combination with a proposed damping matrix. The authors of [10] argue that if an error sample is close to zero then the log-likelihood of that sample approaches infinity as the determinant of the covariance matrix approaches zero. This can be seen in (5.14) since if $\mathbf{e}_i$ is a zero vector then the negative log-likelihood for that error sample is given by

$$
\log|R_i| \tag{5.25}
$$

which goes towards negative infinity as $|R|$ goes towards zero. To solve this problem [10] propose to add a damping matrix E to R which limits the minimum value of the determinant. The estimated covariances is then instead given by

$$\hat{R}_i = R_i + E \tag{5.26}$$

where E is a positive semi-definite matrix.

Similarly as in [10] applying the damping matrix E and L2 regularization to the negative log-likelihood objective one obtains the optimization problem

$$\begin{aligned}\arg\min_{\phi} g(\mathbf{e};\phi) =& \arg\min_{\phi} \sum_{i=1}^{N} \left(\frac{1}{2}\log|\hat{R}_i| + \frac{1}{2}\mathbf{e}_i^T \hat{R}_i^{-1}\mathbf{e}_i\right) + \frac{\lambda}{2}\|\phi\|_2^2 \\ =& \arg\min_{\phi} \sum_{i=1}^{N} \left(-\ell(\hat{R}_i|\mathbf{e}_i)\right) + \frac{\lambda}{2}\|\phi\|_2^2\end{aligned} \tag{5.27}$$

optimized over a data set of error samples and input features $\mathcal{D} = \{\mathbf{e}_i, \mathbf{z}_i | i = 1, 2, \ldots, N\}$.

5.2.5 Optimization

The gradient of the objective function $g(\mathbf{e};\phi)$ with respect to the model parameters has a closed form expression [10]. Before presenting the expression of the gradient a few intermediate definitions are made to make the gradient expression more compact and readable. The same definitions as in [10] are made

$$\begin{aligned}\frac{\partial R(\mathbf{z};\phi)}{\partial w_{i,r}} &= L 1_{ii} L^T D_{ii} \mathbf{z}_r \\ \frac{\partial R(\mathbf{z};\phi)}{\partial v_{ij,r}} &= \left(1_{ij} D L^T + L D 1_{ji}\right) \mathbf{z}_r\end{aligned} \tag{5.28}$$

where $w_{i,r}$ denotes the rth element of the weight vector corresponding to the ith diagonal element of D, $v_{ij,r}$ denotes he rth element of the weight vector corresponding to the ith row and jth column of L, $\mathbf{z}_r$ is the rth element of the input feature vector and lastly 1_{ij} is a matrix which is zero everywhere except at row i and column j where it is 1. Similarly as derived in [10] the partial derivative of the negative log-likelihood for a single error sample $\mathbf{e}_i$ with respect to the kth parameter $\frac{\partial}{\partial \phi_k}\left(-\ell(\hat{R}_i|\mathbf{e}_i)\right)$ is then given by

$$\frac{\partial}{\partial \phi_k}\left(-\ell(\hat{R}_i|\mathbf{e}_i)\right) = \frac{1}{2}\mathrm{tr}\left[\hat{R}_i^{-1}\frac{\partial R_i}{\partial \phi_k}\left(I - \hat{R}_i^{-1}\mathbf{e}_i\mathbf{e}_i^T\right)\right] \tag{5.29}$$

where tr() is the matrix trace and $\frac{\partial R_i}{\partial \phi_k}$ is one of the partial derivatives in (5.28) depending on if ϕ_k is a parameter corresponding to an element in D or L. Considering ϕ as a column vector of the parameters and applying (5.29) to calculate each partial derivative in the gradient, i.e., for each parameter, one may obtain the gradient. The closed form expression of the gradient of $g(\mathbf{e};\phi)$ with respect to the model parameters $\frac{\partial}{\partial\phi}g(\mathbf{e};\phi)$ derived in [10] may thus be described as

$$\frac{\partial}{\partial\phi}g(\mathbf{e};\phi) = \sum_{i=1}^{N}\left[\frac{\partial}{\partial\phi}\left(-\ell(\hat{R}_i|\mathbf{e}_i)\right)\right] + \lambda\phi. \tag{5.30}$$

The interested reader is referred to [10] for a derivation of the gradient.

As one has access to an expression of the objective function gradient with respect to the model parameters the model may be optimized using gradient based optimization methods. In this thesis mini-batch gradient descent is used to optimize the model.

5.2.6 Computational complexity

It is of crucial importance that the covariance estimation may be performed in real-time since it is used in the Kalman filter. If the estimated covariance matrix is $p \times p$ and the number of input features is n then as pointed out in [10] the model has a complexity of $\mathcal{O}(p^2n)$. The complexity of the model thus increases with the size of the error vector samples $\mathbf{e}$ and the number of input features. As the size of the error vectors are relatively small in this thesis and the number of input features may be selected by the model designer this method is appropriate for the project objective.

5.2.7 Model implementation

As mentioned in Section 2.5 the number of measurements may vary in the reference system Kalman filter updates. Additionally, the number of state elements in the state vector for the object prediction step may also differ for different time steps. This problem of missing data is also a problem for PCE since it is not possible to use incomplete error samples in the objective function described in (5.8). We solve this by estimating diagonal covariance matrices as discussed in Section 2.5.3. The elements in the error samples may consequently be used independently of each other and missing data can be discarded without losing useful information. The PCE models consequently estimate scalar variances used to construct diagonal covariance matrices in the object prediction, lane marker update, vehicle update and vehicle heading update.

In the object prediction, vehicle update and vehicle heading update the individual state vector elements and measurement elements from a single time instance are considered to be samples of the same scalar random variable. As such one PCE model which estimates the variance of this scalar random variable is trained for each of these Kalman filter steps. The diagonal covariance matrix is then constructed from the estimated variances and the number of estimated variances depends on the size of the measurement vector or state vector at that time instance.

In the lane marker update, however, the measurement elements are not considered to be samples of the same scalar random variable. The reason for this is that the position of the elements in the measurement vector has a significant meaning. Consequently, multiple PCE models that estimate variances, one for each element in the measurement vector, are trained using the corresponding scalar error sample.

The road prediction step does not suffer from the missing data problem and the error vector samples $\mathbf{e}_i$ may thus be used directly. Nonetheless, a diagonal covariance matrix is used in this Kalman filter step as well since the reference system does not easily allow for dense covariance matrices. Estimating a diagonal covariance matrix is easily imposed by setting the L matrix in the LDL decomposition to the identity matrix.

Lastly, it should be mentioned that categorical input features used in the models were encoded using one-hot encoding discussed in Section 3.1.5. This was done in order to represent these features in a more reasonable way which may improve the learning process of the models.

5.3 Deep Covariance Estimation

Deep Inference for Covariance Estimation (DICE) was proposed in [23] and proposes to use a deep neural network for approximating a function mapping raw measurements to error covariances. As mentioned in [23] a weakness of the parametric method described in Section 5.2 is that it assumes a specific parametric form of the function mapping features to covariances. By using a deep neural network instead the model allows for a more complex nonlinear mapping which does not assume a specific parametric form.

As described by the authors of DICE [23] the method does not require hand-coded input features as covariances are estimated directly from raw measurements. However, in this thesis we have access to hand-coded input features and therefore use these features as model inputs instead of raw measurements. By utilizing features which are already available in the filter the network size may be reduced since the network does not need to learn feature representations from raw measurements. Using a smaller network in turn results in a lower computational complexity which is beneficial as the covariance estimation is used in real-time in the Kalman filter. Another difference of the deep neural network method used in this thesis compared to DICE is that we use a different class of neural networks which will be discussed more in Section 5.3.2.

Because there are several differences between the model used in this thesis and DICE e.g. hand-coded features and network architecture the model used in this thesis is referred to as Deep Covariance Estimation (DeepCE) instead of DICE.

5.3.1 Positive definiteness constraint

The authors of DICE [23] solve the constraint of estimating positive definite matrices in the same way as in [10], by using the LDL decomposition discussed in Section 5.1.2. In this thesis the features were also normalized similarly as in [10] to obtain a more stable optimization.

5.3.2 Model form

Different classes of deep neural networks may be appropriate for a specific task depending on the type of measurements or features used. In [23] a convolutional neural network (CNN) is used which is beneficial if the model inputs have local correlations [43] such as in images which have spatial local correlations. In this thesis a multilayer perceptron (MLP) is used as there are no obvious useful correlations in the input features. A multilayer perceptron is a fully connected neural network. It consists of an input layer, an output layer and a number of hidden layers. An illustration of an MLP is given in Figure 5.1. Furthermore, the type of activation function used in the network in this thesis is the rectified linear unit (ReLU). For more information about ReLU, see [44].

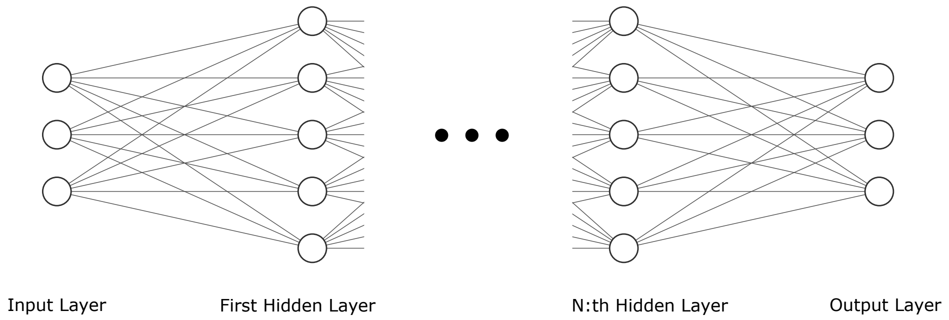


Figure 5.1: Illustration of a multilayer perceptron (MLP). The number of hidden layers and number of neurons in each hidden layer are hyperparameters that need to be chosen.

As mentioned previously, DICE [23] approximates a function mapping from raw measurements to error covariances. However, in this thesis the neural network instead approximates a function mapping from input features to error covariances. Using the notation introduced in Section 5.1.2 the neural network thus approximates the true function $g(\mathbf{z}) = [\boldsymbol{d},\ \boldsymbol{l}]^T$, i.e., from input features $\mathbf{z}$ to LDL decomposition elements $\boldsymbol{d}$ and $\boldsymbol{l}$. The actual function described by the neural network is denoted $f(\mathbf{z})$ and approximates the true function $g(\mathbf{z})$. The aim is thus to achieve $f(\mathbf{z}) \approx g(\mathbf{z})$.

The output layer of the MLP is a linear layer and the output of the neural network may thus be described similarly as in [23]

$$f(\mathbf{z}) = A\gamma + b \tag{5.31}$$

where $\gamma \in \mathbb{R}^n$ is the output from the hidden layers in the neural network, A is a $m \times n$ matrix containing the weight parameters of the linear output layer and b is a column vector of size m containing the bias weights of the linear output layer. The output of the neural network is thus a column vector of size m where m consequently depends on the size of the covariance matrix being estimated since $[\boldsymbol{d},\ \boldsymbol{l}]^T \in \mathbb{R}^m$.

In the same way as for the parametric method the model inputs for DeepCE are not restricted to being discrete and may take any real value. The model inputs are thus given by $\mathbf{z} \in \mathbb{R}^n$ where n is the number of input features.

5.3.3 Objective function

Similarly as in [10] the authors of [23] assume that the error distribution is a zero mean Gaussian with an input dependent covariance. DICE [23] then uses the negative log-likelihood objective discussed in Section 5.1.1 to maximize the likelihood of the errors given the estimated covariances.

5.3.4 Objective regularization

As shown in Section 5.1.3 it is important to use regularization in combination with this objective for estimating covariances. In [23] dropout regularization was used to regularize the network. However, in this work L2 regularization is used instead as it has been shown [45] that smaller networks benefit more from L2 regularization compared to dropout. Applying L2 regularization to the negative log-likelihood objective results in the loss function given in (5.22).

5.3.5 Optimization

As the model function is given by a neural network backpropagation is used to obtain the gradient of the objective with respect to the parameters. The model is then optimized using mini-batch gradient descent with momentum.

5.3.6 Computational complexity

The operations performed to evaluate the neural network consist of multiplications, additions and the max operator in the ReLU [44]. The number of these operations increase with increasing number of inputs, number of hidden layers, number of neurons in each hidden layer and the number of outputs. The number of inputs and outputs is problem dependent while the number of hidden layers and number of neurons may be selected based on a trade-off between performance and computational complexity. By then selecting an appropriately moderate number of hidden layers and neurons the computational complexity may be limited. The computational complexity is furthermore constant in time since the number of parameters in the model is fixed. Hence, DeepCE is a viable approach in real-time systems and therefore also appropriate for the work in this thesis.

5.3.7 Model implementation

Because of the problem discussed in Section 5.2.7 of missing data in the error samples DeepCE also estimates diagonal covariance matrices. The DeepCE models consequently estimate scalar variances similarly to PCE which are used to construct diagonal covariance matrices in the object prediction, lane marker update, vehicle update and vehicle heading update. However, a difference between the implementation of PCE and DeepCE is that even though the measurement elements in the lane marker update are not considered samples of the same scalar random variable only a single DeepCE model is trained. The information contained in the position of the elements in the measurement vector is instead conveyed by adding an input feature to the neural network which describes this position. For the road prediction step the DeepCE model estimates a diagonal covariance matrix directly in the same way and for the same reason described in Section 5.2.7.

Also for the same reason discussed in Section 5.2.7 categorical input features were encoded using one-hot encoding for the DeepCE models.

6
Results

As described in Section 1.4 a heuristic method was available at the start of the thesis which is used to benchmark the proposed solutions. The heuristic method is referred to as Baseline. The methods are evaluated based on two different performance measures on three different test cases with varying road types and environments. The road types considered are highways and country road with varying environmental factors such as clear sky, darkness and rain. These different scenarios are interesting as they potentially have significant impact on the process- and measurement covariances.

6.1 Implementation details

The input features used in PCE and DeepCE for all the Kalman filter models are essentially the same. Some of these selected input features are discussed in Section 3.4. For DCE fewer input features were used because of the scaling problems discussed in Section 4.2.2.

Implementation details specific for each of the modeling methods will now be described in the following subsections.

6.1.1 Discrete covariance estimation

The discrete covariance estimation method was implemented in Matlab and converted to C code using the C code generation utility in Matlab.

6.1.2 Parametric Covariance estimation

The PCE method was implemented in Matlab but converted to C code using the C code generation utility in Matlab. The training parameters were determined by trying different values and choosing the values which resulted in the best performance on the training and development data. The damping matrix E was set to the zero matrix and the L2 regularization parameter was set to $\lambda = 10^{-3}$ during training of all models. The models were trained until convergence using mini-batch gradient descent with a mini-batch size of 1024 for a total of 10 epochs. The first five epochs the learning rate was set to $\eta = 10^{-5}$ and for the last five epochs the learning rate was set to $\eta = 10^{-6}$.

6.1.3 Deep Covariance Estimation

The DeepCE method was implemented in Python using PyTorch but is called from Matlab through an API during runtime.

The neural network architecture was determined by trying different values of the hyperparameters and choosing the values which gave good performance on the training and development sets. The network size was increased until the network started to overfit to the training data. Once the network was able to overfit the L2 regularization parameter was tuned such that the network could be trained until convergence without overfitting. An illustration of the resulting neural network architecture used in this thesis is given in Figure 6.1.

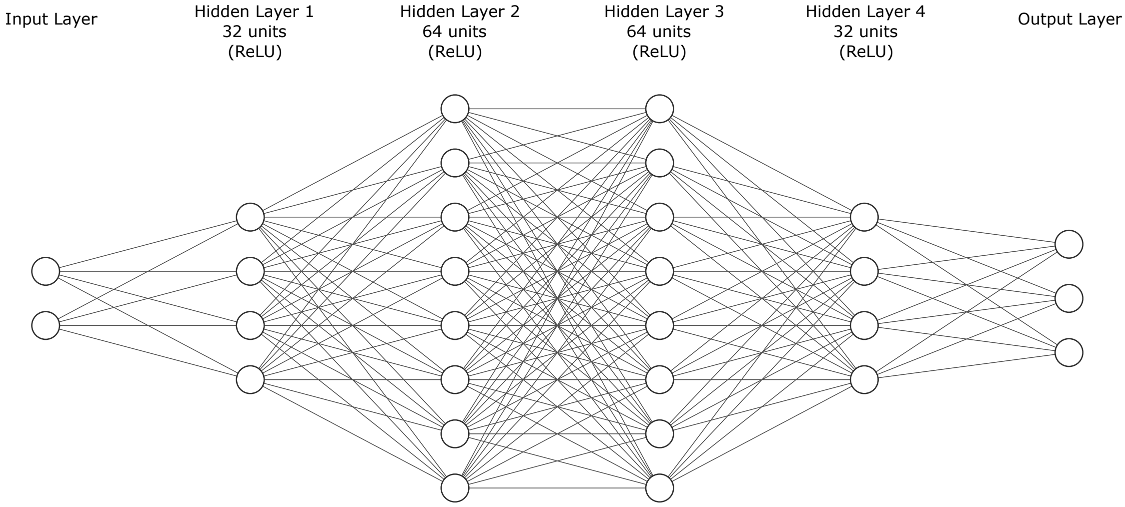


Figure 6.1: Neural network architecture used in the DeepCE model.

The networks were trained using mini-batch gradient descent with momentum where the mini-batch size was set to 1024 and the momentum was set to 0.9 for all models. The L2 regularization was implemented using weight decay and the weight decay parameter was set to $\lambda = 1$ for all models. The networks were trained until the training loss converged. For all Kalman filter models in the reference system except for the lane marker update the covariance estimation models were trained for 160 epochs. For the first 80 epochs the learning rate was set to $\eta = 10^{-6}$ and the last 80 epochs the learning rate was set to $\eta = 10^{-7}$. For the lane marker update the model was trained for 320 epochs where the learning rate was $\eta = 10^{-6}$ for the first 80 epochs and $\eta = 10^{-7}$ for the remaining epochs.

6.2 Test sets

Three different test sets of real-world data are used for evaluation. Below follows a description of each of the sets which describes the scenarios and sizes of the test sets.

1. *HighwayDark*: Highway driving where it gets progressively darker outside as the sun is setting. The data was sampled during a driving session of 61 minutes and data for road geometry estimation evaluation is available for 59 minutes. A picture taken towards the end of the driving session when the sun had set completely is shown in Figure 6.2.

2. *HighwayClearSky*: Highway driving during clear sky. The driving session is 53 minutes long and data for road geometry estimation evaluation is available for 52 minutes. A picture from the driving session is shown in Figure 6.3.

3. *CountryRoadRain*: Driving on a large country road while it is raining heavily. The driving session is in total 50 minutes long and data for road geometry estimation evaluation is available for 24 minutes. It should be noted that the training data set used in this thesis is predominantly from highway scenarios. This test case is still included as it may indicate if models trained on mainly highway data is able to generalize to country roads as well or not. A picture from the driving session is shown in Figure 6.4.

The reason that data for road geometry estimation evaluation is not available for the entirety of the driving sessions is that ground truth data of sufficient quality is not always available.

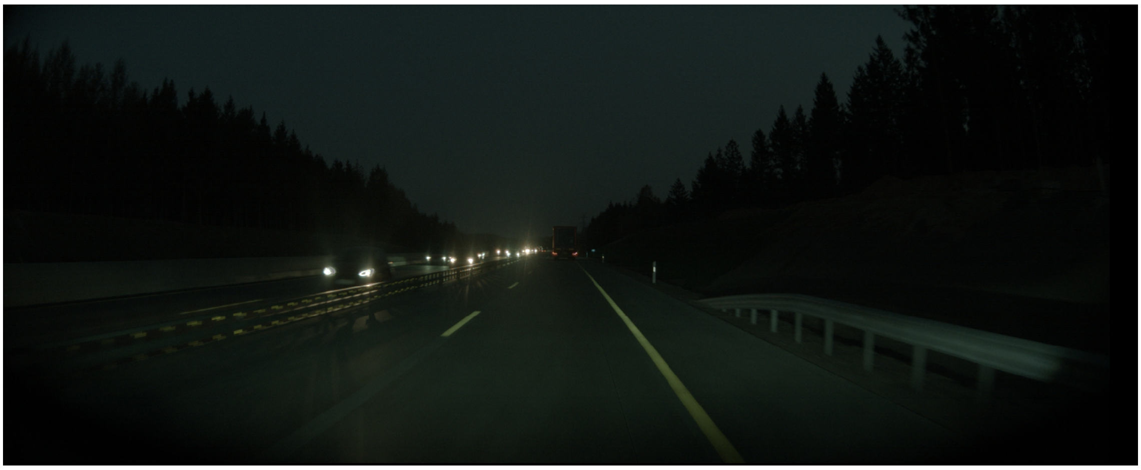


Figure 6.2: Example picture from the test set *HighwayDark*. The picture is taken towards the end of the driving session when the sun had set completely.

Figure 6.3: Example picture from the test set *HighwayClearSky*.

Figure 6.4: Example picture from the test set *CountryRoadRain*.

6.3 Noise negative log-likelihood

Similarly as in [10], a likelihood measure is used to measure the performance of the noise models. The noise negative log-likelihood (NNLL) is the negative log-likelihood of the true error samples given the estimated covariances in the system models, i.e., the process- and measurement covariances. This likelihood is a measure of how well the estimated covariances describe the true error samples and thereby a direct performance measure of the noise model, i.e., the covariance estimation model. Mathematically the NNLL ℓ_{NNLL} is given by

$$\ell_{NNLL} = \sum_{k=1}^{N} \left(\frac{m}{2} \log(2\pi) + \frac{1}{2} \log|R_k| + \frac{1}{2} \mathbf{e}_k^T R_k^{-1} \mathbf{e}_k \right) \tag{6.1}$$

where N is the number of samples in the test data, m is the size of the error vectors,

R_k is the estimated covariance at time k and $\mathbf{e}_k$ is the error vector at time k given by either of the random variable realizations described in (2.6) depending on if the NNLL is calculated for a process- or measurement model.

As explained in Section 2.5 the number of state elements and measurement elements may vary in all Kalman filter models in the reference system except for the road prediction. For the NNLL calculated for these models we solve this by considering the individual error elements and the corresponding variances in the estimated diagonal covariance matrix. In other words, the individual error elements and corresponding variances are stacked into a single sequence of which the sum in (6.1) is taken over. The error $\mathbf{e}_k$ is, thus, a scalar in (6.1) for the object prediction, lane marker update, vehicle update and vehicle heading update. For the road prediction the error vector samples are used directly and therefore not scalars. The results for the three different test sets are shown in Tables 6.1, 6.2 and 6.3 as the normalized average NNLL. As the NNLL is normalized over the different covariance models the best performing covariance estimation model for each Kalman model has a normalized average NNLL of 0.

Normalized average NNLL for test set *HighwayDark*

Kalman model	**DCE**	**PCE**	**DeepCE**	**Baseline**
Road prediction	1	0.0008	0	0.4069
Object prediction	1	0.0026	0	0.1937
Lane marker update	1	0	0.1603	0.0819
Vehicle update	0.0808	0	0.3357	1
Vehicle heading update	1	0.0888	0.0254	0

Table 6.1: Comparison of normalized average NNLL for the different methods in the different Kalman models for test set *HighwayDark*.

Normalized average NNLL for test set *HighwayClearSky*

Kalman model	**DCE**	**PCE**	**DeepCE**	**Baseline**
Road prediction	1	0.0011	0	0.3589
Object prediction	1	0	0.0020	0.0131
Lane marker update	0.2013	0	0.1613	1
Vehicle update	0	0.0369	0.0213	1
Vehicle heading update	1	0.4415	0.4597	0

Table 6.2: Comparison of normalized average NNLL for the different methods in the different Kalman models for test set *HighwayClearSky*.

Normalized average NNLL for test set *CountryRoadRain*

Kalman model	**DCE**	**PCE**	**DeepCE**	**Baseline**
Road prediction	1	0.0011	0	0.3661
Object prediction	1	0.0197	0.0210	0
Lane marker update	1	0	0.8669	0.3109
Vehicle update	0	0.0564	0.2176	1
Vehicle heading update	1	0.0784	0.0722	0

Table 6.3: Comparison of normalized average NNLL for the different methods in the different Kalman models for test set *CountryRoadRain*.

6.4 Root mean square error

The other performance measure used is the root mean square error (RMSE) of the road geometry estimation. The RMSE is interesting as a more consistent process- and measurement noise should result in a better filter performance. Mathematically the RMSE, here denoted as e_{RMSE}, is defined as

$$e_{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \mathbf{e}_k^2} \tag{6.2}$$

where N is the number of samples and $\mathbf{e}_k$ is the error of the estimated position of the road at a certain distance which is given by the Euclidean distance between the estimated point of the road at a certain distance and the closest point of the true road. The estimated position error is thus calculated for several different distances along the road.

The number of samples at different distances can vary because the reference system only produces estimates if it is sufficiently confident at that distance. As a result there are fewer estimates for larger distances. To not present unreliable results the estimates from a certain distance are only considered if there exist at least one minute of produced estimates at that distance. The normalized RMSE of the estimated position of the road at different distances is shown in Figures 6.5, 6.6 and 6.7 for test cases *HighwayDark*, *HighwayClearSky* and *CountryRoadRain* respectively.

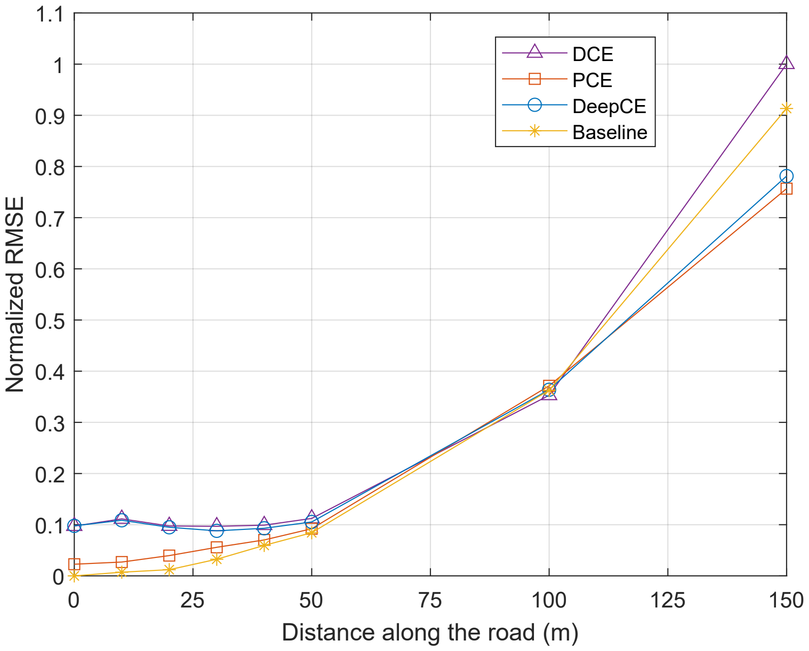


Figure 6.5: Normalized RMSE at different distances along the road for test set *HighwayDark*.

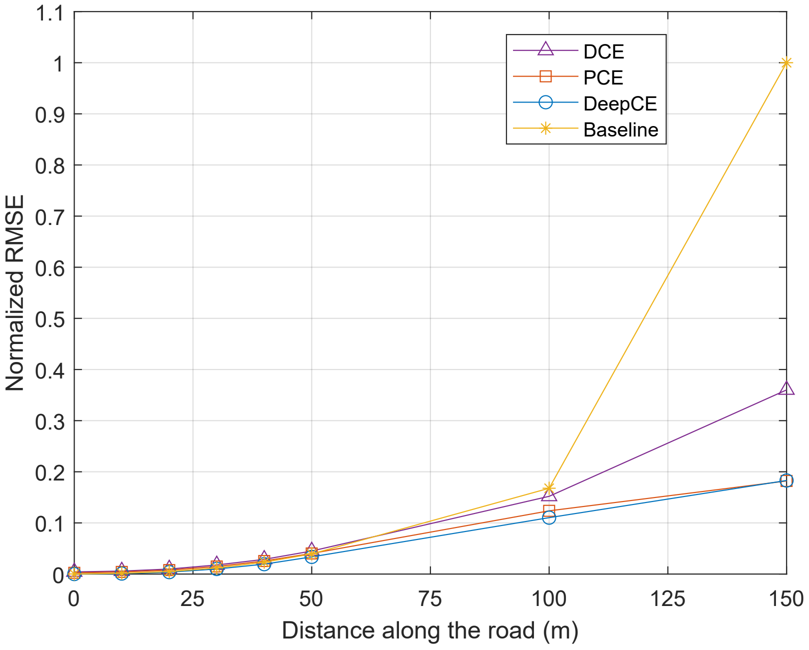


Figure 6.6: Normalized RMSE at different distances along the road for test set *HighwayClearSky*.

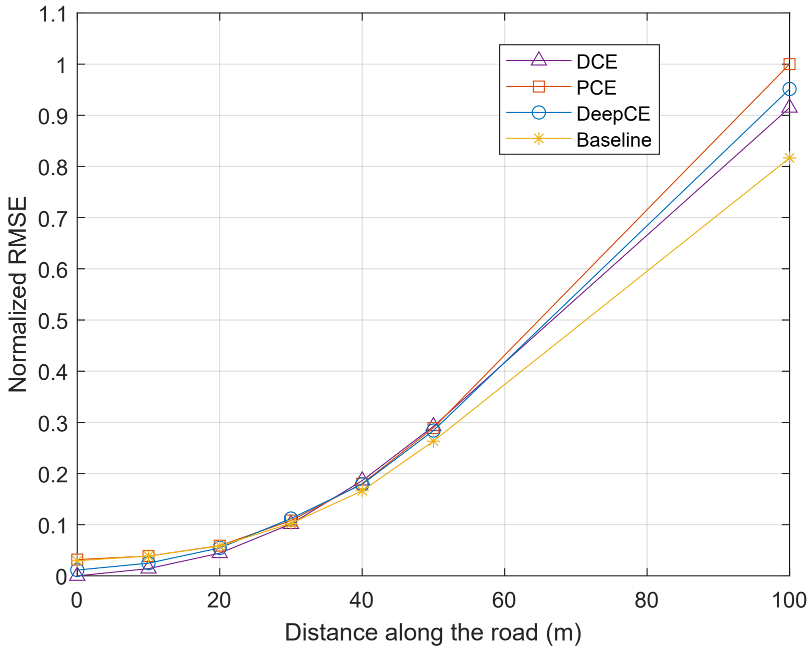


Figure 6.7: Normalized RMSE at different distances along the road for test set *CountryRoadRain*.

7

Discussion

In this chapter the results obtained with regards to NNLL and road geometry estimation RMSE are initially discussed. This is followed by a discussion of potential improvements to different aspects of the thesis work.

7.1 Noise negative log-likelihood

The NNLL results for the three different test cases considered in this thesis can be seen in Tables 6.1, 6.2 and 6.3. For the first test case *HighwayDark*, shown in Table 6.1 PCE performed the best overall but DeepCE also performed well compared to Baseline. PCE performed the best on the vehicle update and vehicle heading update. DeepCE performed the best on the prediction steps, the road and object prediction. However, for the vehicle heading update Baseline performed the best. DCE performed the worst for all Kalman models except for the vehicle update were it was the second best.

In the second test case *HighwayClearSky*, shown in Table 6.2, PCE performed the best overall but the performance of DeepCE was close to PCE. PCE obtained the best results for the object prediction and lane marker update. DeepCE performed the best for the road prediction. DCE performed the best for the vehicle update. Baseline obtained the best result for the vehicle heading update. Lastly it should be mentioned that DCE performed the worst in three of the Kalman filter models.

For the third test case *CountryRoadRain*, shown in Table 6.3, PCE performed the best overall. DeepCE performed slightly better compared to Baseline overall but obtained quite bad results on the lane marker update compared to the other Kalman models for DeepCE. PCE performed the best for the lane marker update. DeepCE obtained the best results for the road prediction. DCE performed the best for the vehicle update. Baseline performed the best for the object prediction and vehicle heading update. Furthermore, DCE obtained the worst performance in four out of five Kalman filter models.

To summarize, over the three different test cases PCE had the best NNLL performance overall but DeepCE also performed better than Baseline in general. The reason that PCE performed the best could be that the simpler parametric form of

PCE compared to DeepCE acted as regularization. In other words it could be that DeepCE overfitted to the training data and in turn did not generalize to the test cases while the simpler parametric form of PCE prevented overfitting. Consequently, by applying more regularization during the training of DeepCE one might expect DeepCE to perform better with regards to NNLL.

Based on the NNLL results it is clear that the covariance estimation models may estimate more consistent covariance matrices, i.e., process- and measurement covariances, compared to Baseline. As these covariances have a significant impact on the Kalman filter estimation covariance one can also reasonably conclude that this results in a more consistent estimation uncertainty compared to Baseline. However, it is important to note that the NNLL is not a direct measure of the Kalman filter estimation uncertainty consistency but instead a measure of the consistency of the estimated covariances in the different Kalman filter steps.

An interesting result to note is that Baseline has the best NNLL performance in the vehicle heading update for all three test cases. A possible reason for this is that it was difficult to find interesting input features for this specific Kalman model. Consequently, the covariance estimation model in the vehicle heading update does not have too many useful input features which could be the reason why Baseline outperforms the covariance estimation methods. By examining the error samples for the vehicle heading update it could also be seen that the error data for this specific Kalman model contained significantly more outliers compared to the other Kalman models. These outliers may in turn be caused by that the measurement model for the vehicle heading update could be more sensitive to noisy ground truth data. In other words when constructing the error data set described in Section 3.2 the vehicle heading update may be more sensitive to noise in the ground truth data resulting in outliers. Obvious outlier samples were removed before training the noise models in the vehicle heading update but the existence of outliers could be an indication that other samples are of lower quality. Hence, the quality of the training data for the noise models in the vehicle heading update may be a reason as to why Baseline consistently performs better for this specific update.

DCE performed the worst overall on all three test cases which could be the result of having to little data in some discrete cases causing the sample covariance to be inaccurate as discussed in Section 4.2.2. One could remedy this by using more training data and more data specifically for the discrete cases where data is lacking. It should also be mentioned that DCE uses fewer input features compared to PCE and DeepCE as we strived to limit the number of discrete cases to solve the problem discussed in Section 4.2.2. However, not having some input features could also be a reason as to why DCE performed the worst regarding NNLL since the missing input features could be useful in estimating accurate covariance matrices. Examining the NNLL results for DCE there is however an exception which is that DCE actually performs quite well specifically for the vehicle update. A plausible reason that DCE performs well on this Kalman model specifically could be that the input features used are exceptionally informative compared to the other Kalman models for DCE. It is also the case that the discrete input features used for the vehicle update for DCE had only a few possible discrete values. This means that the total number of discrete cases for the vehicle update were also few such that each discrete case contained a significant amount of training data. Since this results in more accurate covariance estimates for each discrete case, as more data is used in the sample covariance, this is also a possible reason as to why DCE performs quite well for the vehicle update specifically.

Lastly, it is clear from the NNLL results that different covariance estimation methods performed better or worse for different Kalman models and different test sets. The reason for this is most likely that the different input features found for each Kalman model through feature selection are more or less informative with regards to covariance estimation depending on the test set. It is also the case that the training parameters of the parametric models, PCE and DeepCE, were tuned based on the performance on the development set for the lane marker update specifically. As these parameter values resulted in good performance on the development set also for the other Kalman models we decided not to further fine-tune the training parameters for each Kalman model specifically. Nonetheless, by fine-tuning the training parameters for the noise model in each Kalman model it is possible that one could achieve a more consistent performance over the different Kalman models.

7.2 Root mean square error

The results for road geometry estimation RMSE are shown in Figures 6.5, 6.6 and 6.7 for the three different test cases respectively. In the first test case *HighwayDark*, shown in Figure 6.5, DCE and DeepCE performed significantly worse for short distances compared to Baseline while PCE only performed slightly worse at short distances compared to Baseline. For larger distances PCE and DeepCE both performed better compared to Baseline while DCE performed worse also for larger distances. Overall PCE performed the best out of the methods considered in this thesis and PCE performed similarly to Baseline considering the performance over all distances.

In the first test case the lighting conditions are poor which should have an impact on the camera sensor performance. A crucial reason as to why the covariance estimation models did not consistently perform better than Baseline in the first test case could therefore be that the models did not have an input feature directly related to the lighting conditions. By using an input feature which measures e.g. luminosity one may expect the covariance estimation models to perform even better.

For the second test case *HighwayClearSky*, shown in Figure 6.6, the RMSE performance was similar for all methods including Baseline at short distances. At larger distances all the covariance estimation models performed significantly better compared to Baseline. The parametric methods, PCE and DeepCE, performed similarly overall and also consistently better than DCE. As a whole the best performing models for this test case were the parametric models, PCE and DeepCE.

It is interesting to note that the covariance estimation models performed better compared to Baseline in the second test case where the weather conditions are quite permissive, i.e., the environmental factors should not have too much of an effect on the sensor performance. This shows that when the covariance estimation models do not lack input features which may have a significant impact on e.g. the measurement noise covariance the models may be able to consistently improve the overall road geometry estimation performance.

In the third test case *CountryRoadRain*, shown in Figure 6.7, DCE and DeepCE achieved a slightly lower RMSE at shorter distances compared to Baseline but all models performed worse compared to Baseline at larger distances. Out of the covariance estimation models considered in this thesis DCE performed the best overall and PCE performed the worst overall. However, considering all distances along the road Baseline performed the best overall for this specific test case.

A plausible reason as to why the covariance estimation models performed worse with regards to overall RMSE compared to Baseline for the third test case, *CountryRoadRain*, could be that the road type was a country road while the models were trained predominantly on data from highways. This is plausible since country roads most likely have different road dynamics and conditions related to measurements compared to highways. Examples of this could be that country roads may have a more rapidly changing curvature or that the position of surrounding vehicles may differ compared to highways. By including more data from country roads in the training set one would expect the RMSE performance for this test case to consequently improve. Additionally, in a similar way as for the first test case, in the third test case there is an environmental factor that potentially has a significant impact on the sensor performance, i.e., rain. Since the covariance estimation models do not have an input feature that may indicate if it is raining or not one would expect performance improvements if such an input feature was to be included.

It is interesting to note that even though some covariance estimation models seem to perform better with regards to NNLL for some test sets they do not always consistently perform better regarding the RMSE performance. This can be seen for the third test case, *CountryRoadRain*, by comparing the NNLL results in Table 6.3 with the RMSE results in Figure 6.7. Examining Table 6.3 PCE seems to overall have a better NNLL performance compared to Baseline while in Figure 6.7 PCE performs worse with regards to RMSE compared to Baseline. An explanation for this could be that even though PCE has a better NNLL performance overall it does not achieve a better NNLL compared to Baseline for strictly all five Kalman models. It might then be the case that the Kalman models in which Baseline has a better NNLL performance compared to PCE has a more significant impact with regards to RMSE performance for that specific test scenario. Consequently, Baseline may achieve a better RMSE performance even though e.g. in the third test case PCE performs better overall with regards to NNLL. Another possible reason for the disparity between the NNLL and RMSE results could be that the ground truth state vector data for some state vector elements used to calculate the NNLL performance was of inadequate quality. This is supported by the fact that it could be seen, through examination of the data in plots, that the ground truth state vector data for a few state vector elements was indeed noisy. It might then be the case that the noisy ground truth data caused inaccuracies in the NNLL performance measure such that even though the noise models may perform better compared to Baseline with regards to NNLL, it does not reflect as a consistent RMSE performance increase.

7.3 Improvements discussion

Some interesting features related to environmental factors were not implemented because of time constraints. However, since the methods construct the noise models as heteroscedastic, i.e., feature-dependent, the performance of the models are thus heavily dependent on if the input features are informative. As a result, expanding the features set provides the models with additional information regarding the correlation between the estimation uncertainty and the factors influencing the estimation, measured by the features. One would expect improvements with regards to both estimation performance and the likelihood of the estimated covariances on test cases *HighwayDark* and *CountryRoadRain* if features describing lighting conditions and rain were added.

It is important to mention that improvements of the training data set used in this thesis would most likely lead to a better performance of the models. One possible improvement which was identified was that the ground truth data for a few state vector elements was of lower quality compared to the other state vector elements, i.e., the ground truth data for a few state vector elements was quite noisy. This affects the training data set for the covariance estimation models as the predictions $F_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ and $H_k(\mathbf{x}_k)$ in (2.6) are impacted negatively. By acquiring more accurate ground truth data one would expect to obtain even better results as the training data used to train the models would be of better quality.

Another potential improvement is related to the generation of the training data set. In generating the training data one needs the functions F_k and H_k as described in Section 2.3. In this work we obtained these functions using a tool which ran the reference system with the heuristic covariance method on driving sessions data by storing the function parameters from the run. The predicted true state $F_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ and the noise-free measurement $H_k(\mathbf{x}_k)$ were then calculated by substituting ground truth state vector data into the functions. Consequently, if the function parameters are state-dependent then the stored functions we obtained from running the tool uses state estimates instead of ground truth data to construct F_k and H_k. This approach is not optimal for the vehicle update in the reference system as some of the parameters of the function H_k in the vehicle update are state-dependent. However, as the estimates aim to approximate the ground truth, it should not be a significant problem.

As discussed in Section 7.2 the training data set consisted of data from predominantly highway scenarios. It was also mentioned that this could be the reason why the covariance estimation models overall obtained higher RMSE for the country road test case compared to Baseline. A possible improvement to the work in this thesis is therefore to use more training data from other types of roads but also more training data in general.

An interesting result with regards to NNLL is that a single covariance estimation model did not consistently perform the best for all Kalman models. This can be seen in Tables 6.1, 6.2 and 6.3 since a normalized average NNLL of zero, i.e., the best performing method, is not present in the same column for all Kalman models in any of the tables. Consequently, it could be reasonable to use different covariance estimation models depending on the Kalman model and the input features used in that Kalman model. In some Kalman filter models the relation between the heteroscedstic noise and the input features may be simpler e.g. linear and one may then use simpler models such as DCE and PCE. On the other hand in some Kalman filter models the relation between the heteroscedstic noise and the input features may be more complex e.g. highly non-linear and then it might be more appropriate to use a neural network model such as DeepCE.

8

Conclusion and future work

8.1 Conclusion

In this thesis we presented a framework for heteroscedastic noise estimation in Kalman filtering applied to road geometry estimation. The proposed framework is articulated into two parts: a feature selection part for filtering features based on correlation criteria, and a heteroscedastic noise model. The first model proposed is a straightforward approach that divides the features into discrete cases and maps each case to a specific covariance matrix. Furthermore, two different state-of-the-art approaches are evaluated and modified to fit the work in this thesis, these approaches are a parametric approach and an approach based on deep neural networks.

The methods are evaluated on real-world data corresponding to three different scenarios. The methods are evaluated using a likelihood measure and root mean square error of the road geometry estimation. As shown in Chapter 6, heteroscedastic noise estimation may lead to improvements in both the filter estimation performance, as well as estimation uncertainty consistency for the road geometry estimation application.

The noise models discussed in this thesis provide a way of determining statistically verifiable process- and measurement covariances in the sense that they are determined from actual samples of the noise processes. This is different compared to the common method of viewing the covariances as design parameters tuned based on filter performance which does not necessarily provide statistically verifiable and interpretive covariances.

From the overall results obtained in the three test cases, the best method considered in this thesis for constructing heteroscedastic noise models is PCE. The second best alternative is DeepCE, although it has higher memory and complexity requirements compared to PCE. It is important to highlight how the three methods introduced in this thesis have highly varying requirements. The discrete method DCE has lower computational complexity and lower memory requirements with respect to DeepCE. However, DCE fails to generalize to cases not seen in the training data, which is accomplished by PCE and DeepCE. Since DeepCE is able to describe more complex relations between the input features and the covariance matrices, the choice between PCE and DeepCE can be made based on computational complexity requirements and the input features used in the model.

8.2 Future work

As discussed in this thesis, uncertainty estimation for road geometry depends on a multitude of factors as both the sensors and road geometry dynamics are effected by external factors related to the environment surrounding the vehicle and road conditions. From the test cases considered in this thesis it could be seen that a lack of input features for environmental factors such as lighting conditions and rain may have resulted in lower performance. It is therefore interesting in future work to include these kind of input features and evaluate if they may further improve the noise model performance. An interesting follow-up work could also be to research for relevant additional features which are direct measures of external factors deemed to add useful information to the noise models in the application of road geometry estimation. Furthermore, one could explore different techniques for feature selection in the case where one has a large set of possibly interesting candidate features.

Through examination of the ground truth state vector data in plots it could be seen that the ground truth data for a few state vector elements was noisy. This consequently affects the training data which was used to train the covariance estimation models. In future work one could therefore obtain more accurate ground truth data for these state vector elements as to obtain a training data set of higher quality. Training the covariance estimation models using this data set could then possibly lead to further performance improvements.

Lastly, as the data set used in this thesis was predominantly based on data from highway scenarios, it would be interesting to use more training data from other road types and scenarios. One could then evaluate if the covariance estimation models discussed in this thesis are useful for other road types than highways. It would even be interesting to include more data from highways as more data in general could lead to a performance increase for the covariance estimation models.

Bibliography

[1] L.T. Aarts, J.J.F. Commandeur, R. Welsh, S. Niesen, M. Lerner, P. Thomas, N. Bos, R. J. Davidse, "Study on Serious Road Traffic Injuries in the EU," European Comission, Brussels, Belgium, Oct. 2016.

[2] A. A. Alam, A. Gattami and K. H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, 2010, pp. 306-311.

[3] J. Liu, K. M. Kockelman, and A. Nichols, "Anticipating the emissions impacts of smoother driving by connected and autonomous vehicles, using the MOVES model." *Transportation Research Board 96th Annual Meeting*, 2017.

[4] R. E. Stern et al., "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205-221, Apr. 2018.

[5] Á. F. García-Fernández, L. Hammarstrand, M. Fatemi, and L. Svensson, "Bayesian Road Estimation Using Onboard Sensors," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 4, pp. 1676-1689, Aug 2014.

[6] L. Hammarstrand, M. Fatemi, Á. F. García-Fernández, and L. Svensson, "Long-Range Road Geometry Estimation Using Moving Vehicles and Roadside Observations," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 17, no. 8, pp. 1-15, Feb 2016.

[7] A. Eidehall, J. Pohl, and F. Gustafsson, "Joint road geometry estimation and vehicle tracking," *Control Engineering Practice*, vol. 15, no. 12, pp. 1484-1494, 2007.

[8] C. Lundquist and T. B. Schön, "Joint Ego-Motion and Road Geometry Estimation," *Information Fusion*, vol. 12, no. 4, pp. 253-263, Oct 2011.

[9] W. Ding, J. Wang, C. Rizos, "Improving Adaptive Kalman Estimation in GPS/INS Integration," *The Journal of Navigation*, vol. 60, no. 3, pp. 517-529, Sep. 2007.

[10] H. Hu and G. Kantor, "Parametric covariance prediction for heteroscedastic noise," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 3052-3057.

[11] R. K. Mehra, "On the Identification of Variances and Adaptive Kalman Filtering," IEEE Transactions on Automatic Control, vol. AC-15, no. 2, Apr 1970, pp. 175-184.

[12] A. H. Mohamed and K. P. Schwarz, "Adaptive Kalman Filtering for INS/GPS.", *Journal of Geodesy 73.4*, pp. 193-203, 1999.

[13] A. Censi, "An accurate closed-form estimate of ICP's covariance," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Roma, 2007, pp. 3167-3172, doi: 10.1109/ROBOT.2007.363961.

[14] E. B. Olson, "Real-time correlative scan matching," *2009 IEEE International Conference on Robotics and Automation*, Kobe, 2009, pp. 4387-4393, doi: 10.1109/ROBOT.2009.5152375.

[15] M. Pupilli and A. Calway, "Real-Time Visual SLAM with Resilience to Erratic Motion," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, 2006, pp. 1244-1249, doi: 10.1109/CVPR.2006.240.

[16] H. Hu and G. Kantor, "Parametric covariance prediction for heteroscedastic noise," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 3052-3057, doi: 10.1109/IROS.2015.7353798.

[17] M. Pourahmadi, "Joint Mean-Covariance Models with Applications to Longitudinal Data: Unconstrained Parameterisation." *Biometrika*, vol. 86, no. 3, 1999, pp. 677–690.

[18] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression", In Proceedings of the 24th international conference on Machine learning (ICML '07), Association for Computing Machinery, New York, NY, USA, pp. 393–400, 2007, doi: https://doi.org/10.1145/1273496.1273546.

[19] W. Vega-Brown, A. Bachrach, A. Bry, J. Kelly and N. Roy, "CELLO: A fast algorithm for Covariance Estimation," 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, 2013, pp. 3160-3167.

[20] H. Coskun, F. Achilles, R. DiPietro, N. Navab and F. Tombari, "Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 5525-5533, doi: 10.1109/ICCV.2017.589.

[21] R. Jonschkowski, D. Rastogi, and O. Brock, "Differentiable Particle Filters:

End-to-End Learning with Algorithmic Priors.", *Robotics: Science and Systems*, ArXiv, abs/1805.11122, May 2018.

[22] A. Kendall and Y. Gal, “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?”,
textitAdvances in Neural Information Processing Systems pp. 5574-5584, 2017.

[23] K. Liu, K. Ok, W. Vega-Brown and N. Roy, "Deep Inference for Covariance Estimation: Learning Gaussian Noise Models for State Estimation," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 1436-1443.

[24] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995, pp. 41-95.

[25] J. A. Gubner, "Random Vectors," in *Probability and random processes for electrical and computer engineers*, Cambridge University Press, 2006, ch. 7, pp. 223-224.

[26] R. J. A. Little, "A test of missing completely at random for multivariate data with missing values," *Journal of the American statistical Association*, vol. 83, no. 404, pp. 1198-1202, 1988.

[27] L. Wasserman, "Expectation," in *All of statistics: a concise course in statistical inference*, Springer Science & Business Media, 2013, ch. 3, pp. 52-54.

[28] T. C. Haas, "Statistical assessment of spatio-temporal pollutant trends and meteorological transport models," *Atmospheric Environment*, vol. 32, no. 11, pp. 1865-1879, 1998.

[29] Draper, Norman R., and Harry Smith. Applied Regression Analysis, John Wiley & Sons, Incorporated, 1998.

[30] Baltagi, B. H. (2011). Econometrics. Springer Berlin Heidelberg.

[31] S. Chatterjee and J. S. Simonoff, Handbook of Regression Analysis, John Wiley & Sons, Incorporated, 2012.

[32] M. Verbeek, "A Guide to Modern Econometrics", 2nd edition. Chichester, England: John Wiley & Sons, 2008.

[33] W. H. Greene, "Econometric Analysis", fifth edition, Prentice Hall, Upper Saddle River, New Jersey 07458, Jul 2002.

[34] Heumann, C., Schomaker, M., & Shalabh. (2016). Introduction to Statistics and Data Analysis. Springer International Publishing.

[35] Rodríguez, G. (2007). Lecture Notes on Generalized Linear Models. Available: `http://data.princeton.edu/wws509/notes/`.

[36] M. L. Bermingham, et al. “Application of high-dimensional feature selection: evaluation for genomic prediction in man,” *Scientific reports*, vol. 5:10312, May 2015.

[37] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research 3*, Mar 2003, pp. 1157-1182.

[38] L. Rutkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, J. M. Zurada, "Artificial Intelligence and Soft Computing, Part I", 10th International Conference, ICAISC 2010, Zakopane, Poland, June13-17, 2010, Part I, pp. 290-300.

[39] A. Jović, K. Brkić and N. Bogunović, "A review of feature selection methods with applications," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2015, pp. 1200-1205, doi: 10.1109/MIPRO.2015.7160458.

[40] N. Sánchez-Maroño, A. Alonso-Betanzos, M. Tombilla-Sanromán, "Filter Methods for Feature Selection – A Comparative Study", In: H. Yin, P. Tino, E. Corchado, W. Byrne, X. Yao, (eds) "Intelligent Data Engineering and Automated Learning - IDEAL 2007", IDEAL 2007, *Lecture Notes in Computer Science*, vol 4881, Springer, Berlin, Heidelberg, 2007.

[41] R. A. Johnson and D. W. Wichern, "Applied Multivariate Statistical Analysis.", Englewood Cliffs, N.J.: Prentice Hall, 1992.

[42] M. Jordan. (2009). The Multivariate Gaussian [Online]. Available: `https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter13.pdf`.

[43] R. Kozma, et al. eds. "Deep Learning Approaches to Electrophysiological Multivariate Time-Series Analysis," in *Artificial Intelligence in the Age of Neural networks and Brain computing.* Academic Press, 2018, ch. 11.

[44] S. Skansi, "Convolutional Neural Networks," in *Introduction to Deep Learning: from logical calculus to artificial intelligence*, Springer, 2018, ch. 6, pp. 121-124.

[45] E. Phaisangittisagul, "An Analysis of the Regularization Between L2 and Dropout in Single Hidden Layer Neural Network," 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), Bangkok, 2016, pp. 174-179, doi: 10.1109/ISMS.2016.14.