Speed and yaw rate estimation in autonomous vehicles using Doppler radar measurements

MARC SIGONIUS
Speed and yaw rate estimation in autonomous vehicles using Doppler radar measurements

MARC SIGONIUS

Master Thesis at EECS
Industrial Supervisor: Patricio E. Valenzuela
Academic Supervisor: Matias I. Müller
Examiner: Cristian R. Rojas

TRITA-EECS-EX-2018:265
Abstract

One of the key elements for model-based autonomous driving is the ability of the vehicle to accurately predict its own motion. Knowledge about the motion can then be widely used, e.g. for localization, planning and control.

This thesis presents an algorithm that estimates the velocity and the yaw rate based on Doppler radar measurements. This system uses an Unscented Kalman filter to extract the motion of the vehicle from multiple Doppler radar sensors mounted on the vehicle. The estimation of these quantities is shown to be critically dependent on outlier detection and the vehicle’s center of rotation. This work presents a framework for detecting dynamical objects, as well as estimating the center of rotation of the vehicle effectively.

In tests, the proposed implementation shows better root-mean squared error performance than the current employed algorithm by 28.8% and 22.4% for velocity and yaw rate, respectively.
Referat

Estimering av hastighet och vinkelhastighet för autonoma fordon baserat på radarmätningar

Ett krav för att kunna köra ett modellbaserat autonomt fordon är att fordonet kan prediktera sin rörelse. Dess rörelse kan vidare användas i många olika applikationer, till exempel lokalisering, planering och reglering.


Den föreslagna algoritm har visat sig ge lägre RMSE-värden med 28,8% för hastigheten och 22,4% för vinkelhastigheten jämfört med nuvarande algoritm.
Contents

1 Introduction ................................................. 1
  1.1 Purpose and goals ........................................ 2
  1.2 Problem formulation .................................... 2
  1.3 Thesis outline .......................................... 2

2 Background Theory ........................................ 3
  2.1 Related work ............................................ 4
    2.1.1 Velocity and yaw rate estimation based on Doppler radar . . 4
    2.1.2 Unscented Kalman filter for radar based ego-estimation . . . 5
  2.2 Current velocity and yaw rate estimation based on Doppler radar . . . 6
  2.3 System and filtering theory ................................ 6
    2.3.1 Kalman filter ...................................... 7
    2.3.2 Extension to non-linear models .................... 8
    2.3.3 Extended Kalman filter (EKF) ..................... 8
    2.3.4 Unscented Kalman filter (UKF) .................. 9
    2.3.5 Filtering with errors-in-variables using UKF .............. 11
  2.4 Outlier detection ....................................... 13
    2.4.1 Definition ...................................... 13
    2.4.2 Threshold-based outlier detection .................. 13
  2.5 Sensor theory .......................................... 16
    2.5.1 Doppler radar .................................... 16
    2.5.2 Aliasing in Doppler radars ....................... 17

3 Method ......................................................... 19
  3.1 Vehicle coordinate system ................................. 19
  3.2 Vehicle setup ........................................... 19
  3.3 Models for the vehicle motion ............................ 20
    3.3.1 Constant velocity model .......................... 20
    3.3.2 Constant acceleration model ....................... 21
  3.4 Measurement model ...................................... 21
  3.5 Center of rotation ...................................... 23
  3.6 Architecture of the estimation procedure ............... 24
    3.6.1 Preprocessing .................................... 24
CONTENTS

3.6.2 Outlier detection .............................................. 25
3.6.3 Main filter ...................................................... 27
3.6.4 Post processing ................................................. 30

3.7 Process and measurement covariances .......................... 30
3.7.1 Doppler velocity measurement ................................. 31
3.7.2 Angle measurement ........................................... 31
3.7.3 Model velocity and yaw rate noise ............................ 32
3.7.4 Lateral and angular acceleration noise ......................... 32

3.8 Tuning parameters of the unscented Kalman filter .......... 32
3.9 Observability of the states ....................................... 33

4 Testing .............................................................. 35
4.1 Performance assessment .......................................... 35
4.1.1 "Ground truth" .................................................. 35
4.1.2 Root-mean-square error (RMSE) ............................... 35
4.1.3 Maximum absolute error (MAE) ............................... 36
4.1.4 Autocorrelation ................................................ 36
4.1.5 Cross-covariance .............................................. 36
4.1.6 Histogram fit of Gaussian distributed data .................. 37
4.1.7 P-P and Q-Q plot ............................................. 37
4.2 Overfitting and cross-validation .................................. 37
4.3 Test scenarios ...................................................... 38
4.3.1 Collection of data for filter parameter tuning ............... 38
4.3.2 Scenarios to evaluate performance ............................ 39
4.4 Time delay compensation .......................................... 39

5 Results .................................................................. 41
5.1 Ground truth prefiltering .......................................... 41
5.2 Filters implemented to compare with the proposed system . 42
5.3 Tuning parameters .................................................. 43
5.3.1 Estimation of the measurement noise ......................... 43
5.3.2 Estimation of process noise ................................... 43
5.3.3 Outlier threshold .............................................. 43
5.4 Model evaluation .................................................... 44
5.5 Outlier detection performance ..................................... 47
5.6 Center of rotation .................................................. 49
5.7 Unscented Kalman filter performance ............................. 50
5.8 Overall performance .............................................. 51

6 Discussion ............................................................ 55
6.1 Performance of the filtering algorithm ......................... 55
6.2 Tuning ............................................................... 56
6.3 Prediction and measurement models .............................. 56
6.4 Outlier detection .................................................... 57
CONTENTS

6.5  Center of rotation ........................................... 58
6.6  Estimation method ........................................... 59

7  Conclusion ......................................................... 60
   7.1  Thesis summary ............................................... 60
   7.2  Future work .................................................. 60

Bibliography ....................................................... 62
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KF</td>
<td>Kalman filter</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman filter</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman filter</td>
</tr>
<tr>
<td>UT</td>
<td>Unscented Transform</td>
</tr>
<tr>
<td>AUKF</td>
<td>Augmented Unscented Kalman filter</td>
</tr>
<tr>
<td>GRV</td>
<td>Gaussian Random Variable</td>
</tr>
<tr>
<td>RANSAC</td>
<td>Random Sampling Consensus</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>CTRV</td>
<td>Constant Turn Rate Velocity</td>
</tr>
<tr>
<td>CTRA</td>
<td>Constant Turn Rate Acceleration</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>MAE</td>
<td>Maximum Absolute Error</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Autonomous driving is currently one of the most exciting and growing areas within the transport sector. As a step towards autonomous driving, manufacturers provide software services and functions that can help drivers to make their travel easier and more comfortable. Among its features, it can prevent traffic accidents caused by human mistakes. But autonomous vehicles can also create better comfort for the society. Autonomous buses can transport people, trucks can deliver goods during uncomfortable working hours and autonomous cars can make it unnecessary to own a car in the future.

The vehicles are equipped with more and more computational power, sensors and advanced software functions to create safety, comfort, sustainability and economic benefits. But increasing the system’s complexity also introduces new challenges, where the current and future technology can handle more complex computations, such computations that are needed for autonomous driving.

One of the goals of autonomous driving is that the vehicle must operate in all different environments and situations. Wherever the vehicle is positioned and in whatever environment, an autonomous vehicle should be able to handle the situation. The situations in which the vehicle should perform well can be limited to some specific environments, such as driving in a mine or at highways, but the vehicle should always be able to drive safely and handle the environment even if it accidentally is placed outside the specified environment, e.g. if a mining vehicle is forced out of the mine to prevent an accident. In such situations, an underlying parametric physical model can be crucial to ensure that the system is robust.

A key element needed in autonomous vehicles is the ability to accurately predict and track objects in its surroundings, as well as in determining its own position. To achieve this goal, the vehicle uses information from local sensors to estimate the required dynamics. One of these sources of information is available in the radar measurements of the vehicle, which can be employed to estimate the speed and yaw rate of the host vehicle. These quantities can for example be used as a state feedback observer for control of the vehicle, prediction of the position for route planning and localization, object tracking and detection of obstacles. With autonomous vehicles,
the demand for accuracy and precision increases even more.

In this context, the thesis explores the use of filtering algorithms and proposes a method for estimating the speed and yaw rate of the host vehicle based on radar detections. The work has been carried out as a collaboration between Scania CV AB and KTH Royal Institute of Technology within the scope of a master thesis in Systems, Control & Robotics.

1.1 Purpose and goals

The main goals are:

- To study available methods for estimating the speed and yaw rate of the host vehicle based on radar detections, and determine the most suitable for the current context,
- to identify and implement potential improvements in the dynamical models used for estimation,
- to examine the cases where the estimation of speed and yaw rate from radar detections are reliable and propose how to detect them in real time, and
- to identify and implement potential improvements in outlier detection.

1.2 Problem formulation

Using radar detections, is it possible to improve the current speed and yaw rate estimation by improving both the algorithm and the non-linear model?

1.3 Thesis outline

This thesis has the following outline: Chapter 2 introduces the background theory, related work and existing implementations, and motivates the choice of algorithms. Chapter 3 describes the implementation of the method used for velocity and yaw rate estimation. Chapter 4 introduces how the performance is measured and the system is tested. Chapter 5 describes the results and in Chapter 6 a discussion about the implementation and the results is done. Finally, Chapter 7 presents conclusions and future work.
Chapter 2

Background Theory

The first velocity estimation was used to provide a driver with information about how fast the vehicle moves. The aim of such system was to help the driver to perform manual control of the vehicle speed to stay within speed limits and to keep similar velocity during longer drives. The precision of this system would not be needed to be very high, since the manual control normally will not be able to control the speed with a very high precision. However, in order to be a measurement that the driver can understand and trust, some accuracy is indeed needed [1]. For this purpose, the speedometers in the vehicles are accurate enough today. Even if speedometers are still used today, and included in every produced vehicle, it is not the type of application that mainly drives the research forward towards a more accurate estimation.

Today, velocity estimation is used for many different purposes in the vehicle such as gear-shifting, automatic cruise control and in safety systems. These systems can require more accurate estimates of the vehicle speed in order to make the vehicle work appropriately.

Today and in the future, the velocity and the yaw rate are quantities that are and will be widely used in autonomous driving. They contain important information that partly explains how a vehicle moves and behaves. The quantities can for example be used as a state feedback observer for vehicle control, prediction of the position for route planning and localization, object tracking and detection of obstacles. With autonomous vehicles, the demand for accuracy and precision increases even more.

There are several different sensors that can estimate the velocity and/or yaw rate. They can be local sensors, measuring a physical quantity regarding a particular component, such as wheels, or sensors that are based on measurements towards global landmarks. Three types of sensors whose measurements can be used for velocity and yaw rate estimation are wheel encoders, accelerometers and Doppler radar sensors. The wheel encoders, which are typically used for speedometers, can deliver an estimate with high precision of the speed in many situations, but can suffer from systematic and random errors [2]. Systematic errors occur for example due
to unequal wheel diameters or uncertainties about the exact wheelbase, while non-
systematic errors are due to wheel slipage, bumps or cracks [2]. Yet, accelerometers
suffer from low signal-to-noise ratio (SNR) at low speeds and from extensive drift [3].
With this in mind, an estimation based on Doppler radar can be beneficial as a complement. An estimate based on detections from Doppler radar sensors will also be independent of any other sensors, thus it can be used for redundancy of the system. At low speeds, the Doppler radar sensors also have shown good performance in previous work [4].

Along the remainder of this chapter, the current velocity and yaw rate estimation theory will be described. The background theory of sensors and filter techniques that will be used for the proposed algorithm will also be introduced.

2.1 Related work

2.1.1 Velocity and yaw rate estimation based on Doppler radar

During the last decades, considerable research effort has been made for using Doppler radars for ego-estimation of velocity and yaw-rate. In 1993, Kleinhempel [5] introduced a Doppler speedometer, showing excellent system performance even under adverse environmental conditions.

In 2006, Hantsch and Mentzel presented a first approach for ego-motion estimation using Doppler radars [6]. The radar was mounted underneath the vehicle pointing towards the road surface. With this setup, the quality of the velocity estimation is dependent on the road quality and the unique positioning of the radar does not make it suitable for usage in other applications. Later, in 2006, Fölster and Rohling [7] proposed an approach of velocity estimation with a single Doppler radar mounted in lateral direction. The algorithm that was proposed assumed that the azimuth angle was small and that the distance to objects were less than 20 meters. These assumptions were done to get well-conditioned equations for estimating the desired variables.

More recently, in Kellner et. al. [2] a system for instantaneous ego-motion estimation was presented. This approach is a three step procedure. The first step is outlier detection and detection of non-stationary objects. This step is done by random sample consensus (RANSAC) [8], an iterative process dropping out data-points not fitting regressions obtained with random subsets of samples. Once outliers are excluded, a velocity profile is extracted using least squares. In the last step, the ego-estimation is done using a single-track model with the Ackermann condition based on the velocity profile that was extracted in the second step.

A limitation of the least squares estimation of the velocity is that it is not bias-
free due to noise in the azimuth angle measurements. This issue was discussed in [2] and the same authors proposed a bias-free solution to this in [9]. The setup with a three step procedure was similar to [2], but instead of using a least squares estimator, the authors proposed a velocity profile extraction based on orthogonal regression analysis, also called errors-in-variables regression. This solution handles the errors-
in-variables issue that is neglected by least squares and simulations support the conclusion that the proposed algorithm is, in fact, bias-free. In [10], this algorithm is evaluated and extended to be compatible with multiple Doppler radars. The results show that measurements from multiple radars improve the accuracy of the estimation [9, 10].

In 2015, Kellner et al. [11] proposed a new algorithm to estimate velocity based on both spatial radar detections and a Doppler-based metric. The spatial metric was based on mixture of Gaussians, where each detection is treated as a Gaussian random variable. Two measurements are then matched by an $\ell_2$ metric to estimate the velocity of the vehicle. The Doppler measurements are compared with a prediction of the velocity, and then the difference and its estimated variance are used in an optimization problem that matches the mixtures of Gaussians with the Doppler measurements. The algorithm showed improvements in accuracy and bias compared to previously proposed algorithms.

A similar approach to [11] was used in [12], where instead of using a mixture of Gaussians with all single detections as in [11], $k$-means clustering was used to obtain a sparse Gaussian mixture model. In [13] a comprehensive description and evaluation of this method is presented, where an improvement of accuracy is shown, but with a higher computational cost compared to previous methods.

2.1.2 Unscented Kalman filter for radar based ego-estimation

The Unscented Kalman Filter (UKF) is a filter that was introduced to handle non-linearities in a dynamical model, originally presented in [14]. This filter is often compared to the more commonly used Extended Kalman Filter (EKF), in which the state distribution is approximated by a Gaussian Random Variable (GRV) being propagated through a first order linearization of the system dynamics. Nevertheless, this can lead to sub-optimal performance and sometimes divergence of the filter [15]. The UKF addresses this problem by an unscented transform, whose formal definition is postponed to Section 2.3.4. The state distribution is still approximated by a GRV, but this time the GRV is represented by a minimal set of carefully chosen sigma-points. These sigma-points capture the true mean and covariance of the GRV. When these sigma-points are propagated through the non-linear system, the posterior mean and covariance are captured up to the third order of the Taylor series expansion of any non-linear system. Compared to the EKF, that only captures dynamics from first order Taylor expansions, the UKF has demonstrated performance gains in state estimation for non-linear systems [14, 15].

In [16], the effect of taking noise into account on model parameters is investigated using radar detections for estimating polynomial structures. The results show that a UKF can be used for noisy-model parameter estimation and that the estimation becomes more accurate when noise is accounted for.

In [17], a UKF-based approach using Doppler radar measurements for improvement of the position estimate is presented. This implementation uses a Direct UKF, where the measurements are used directly in the filter, without any pre-processing.
The result is based on Monte Carlo simulations and shows that the UKF performs well when the measurements have high accuracy and do not have strong correlation with the range measurements.

2.2 Current velocity and yaw rate estimation based on Doppler radar

The velocity and yaw rate estimation with Doppler radar sensors is currently based on [2]. However, in contrast to [2], the outlier detection is not based on RANSAC, but on $\chi^2$-statistics, whose formal definition is postponed to Section 2.4.2. The estimation of velocity and yaw rate is done with a least squares optimization and the results from the least squares optimization are then fed into a Kalman filter that updates the states.

As mentioned in Section 2.1, the least squares optimization is biased due to neglected uncertainty in measurements. This has also been detected and identified as an issue with the current implementation prior to this thesis. The current implementation is also a projection/optimization in multiple steps as shown in Figure 2.1. This means that the optimal projection in one step might not be the optimal in the next step, since each step is making an optimal projection based on the available information. The least squares estimation is a dimension reduction from $n$ measurements to one estimate of the velocity and one of the yaw rate. From the dimension reduction, it follows that some information is lost in the projection, which the Kalman filter needs to make an optimal prediction.

2.3 System and filtering theory

This section will describe the filtering theory and the system architecture.
CHAPTER 2. BACKGROUND THEORY

2.3.1 Kalman filter

The Kalman filter is an optimal estimator with respect to any quadratic function of estimation error [18]. The filter operates with a state space model which, together with available sensor data, generates an estimate of the current state.

The standard Kalman Filter’s prediction and measurement models are linear and can be described as

\[
\begin{align*}
\dot{x}_{k+1|k} &= A_k x_{k|k} + B_k u_k + v_k, \\
z_k &= H_k x_{k|k} + w_k.
\end{align*}
\]

The processes \(\{v_k\} \in \mathbb{R}^n\) and \(\{w_k\} \in \mathbb{R}^b\) are both assumed to be zero-mean and white with covariance matrices \(Q \in \mathbb{R}^{n \times n}\) and \(R \in \mathbb{R}^{b \times b}\). Thus, its best estimates are their means, which are zero. The prediction and measurement models for the Kalman filter are then described by

\[
\begin{align*}
\hat{x}_{k+1|k} &= A_k \hat{x}_{k|k} + B_k u_k, \\
\hat{z}_k &= H_k \hat{x}_{k+1|k}.
\end{align*}
\]

In (2.3) the prediction model is described. The current predicted state, \(\hat{x}_{k+1|k} \in \mathbb{R}^n\) is assumed to be described as a linear function of the previous state, \(x_{k|k} \in \mathbb{R}^n\), and some known input \(u_k \in \mathbb{R}^m\). The linear function is composed by the known matrices \(A_k \in \mathbb{R}^{n \times n}\) and \(B_k \in \mathbb{R}^{n \times m}\), which describe the dynamics of the state that is estimated. In (2.4), the matrix \(H_k \in \mathbb{R}^{b \times n}\) describes the mapping from the predicted state to \(b\) measurements from sensors.

The algorithm is an iterative process with two steps: a prediction step and a measurement update step.

1. **Prediction Step:**

\[
\begin{align*}
\hat{x}_{k+1|k} &= A_k \hat{x}_{k|k} + B_k u_k, \\
P_{k+1|k} &= A_k P_{k|k} A_k^T + Q,
\end{align*}
\]

2. **Measurement Update Step:**

\[
\begin{align*}
K_k &= P_{k+1|k} H_k^T (H_k P_{k+1|k} H_k^T + R)^{-1}, \\
\hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_k (z_{k+1} - H_k \hat{x}_{k+1|k}), \\
P_{k+1|k+1} &= (I - K_k H_k) P_{k+1|k},
\end{align*}
\]

where \(z_{k+1} \in \mathbb{R}^b\) is the measured value and \(I \in \mathbb{R}^{n \times n}\) is the identity matrix.

In the prediction step, the previous estimates of the state and the covariance matrix are propagated through a model that exploits the knowledge about the system dynamics. This prediction is a projection forward in time to obtain the a priori estimates for the next time step. The measurement update step is responsible for the feedback from measurements, i.e. incorporating the latest measurement that describes the system to the a priori estimate to obtain the a posteriori estimate [19].
2.3.2 Extension to non-linear models

The optimality of the Kalman filter is restricted only to linear system dynamics. This is a limitation since most dynamical systems are non-linear. A typical extension of (2.3) and (2.4) is a model described by

\[
\begin{align*}
    x_{k+1|k} &= f(x_k|k, u_k, v_k) \\
    z_{k+1} &= h(x_{k+1|k}, w_k)
\end{align*}
\]

(2.10) (2.11)

where \(f\) and \(h\) are known continuous functions and \(\{v_k\}\) and \(\{w_k\}\) still are assumed to be zero-mean white with covariances \(Q\) and \(R\). In this case, two different extensions of the KF will be considered, the extended Kalman filter and the unscented Kalman filter.

2.3.3 Extended Kalman filter (EKF)

The extended Kalman filter (EKF) is an estimator that linearizes the system dynamics around the current state. This results in a linear model that can be used to estimate the current state based on the previous state and some measurements. In practice, the best estimate of white noise is its mean value. Since both \(v_k\) and \(w_k\) are zero-mean, their expected values are 0. Thus, a linearization around \(\hat{x}_{k+1|k} = f(\hat{x}_k|k, 0)\) \(\hat{z}_{k+1} = h(\hat{x}_{k+1|k}, 0)\)

(2.12) (2.13)

can be used to achieve a linearized model. The linearization is done by the Jacobians

\[
\begin{align*}
    A_k &= \frac{\partial f(s, u_k, 0)}{\partial s} \bigg|_{\hat{x}_{k|k}} \\
    H_k &= \frac{\partial h(s, 0)}{\partial s} \bigg|_{\hat{x}_{k+1|k}}.
\end{align*}
\]

(2.14) (2.15)

The iterative process is then described as follows:

1. **Prediction Step:**

\[
\begin{align*}
    \hat{x}_{k+1|k} &= f(\hat{x}_k|k, u_k, 0) \\
    P_{k+1|k} &= A_k P_{k|k} A_k^T + Q.
\end{align*}
\]

(2.16) (2.17)

2. **Measurement Update Step:**

\[
\begin{align*}
    K_k &= P_{k+1|k} H_k^T (H_k P_{k+1|k} H_k^T + R)^{-1} \\
    \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_k (z_{k+1} - h(\hat{x}_{k+1|k}, 0)) \\
    P_{k+1|k+1} &= (I - K_k H_k) P_{k+1|k}.
\end{align*}
\]

(2.18) (2.19) (2.20)
A fundamental flaw of the EKF is that the distributions of the various random variables are no longer Gaussian distributed after the non-linear transformation. The EKF can therefore be seen as an ad hoc state estimator that only approximates the optimality of Bayes’ rule by linearization \[19\].

### 2.3.4 Unscented Kalman filter (UKF)

The Unscented Kalman Filter (UKF) was introduced to handle non-linearities in the process and measurement models of the Kalman filter without any linearization. The filter was originally presented in \[14\] by Julier and Uhlmann. This filter is often compared to the more commonly used EKF, in which the state distribution is approximated by a GRV which is propagated through a first order linearization of the system dynamics. The UKF addresses the problem by usage of an unscented transform (UT). The UT is based on the fact that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function \[20\].

The state distribution is still approximated by a GRV, but this time the GRV is represented by a minimal set of carefully chosen sigma-points. These sigma-points capture the true mean and covariance of the GRV. When these \(\sigma\)-points are propagated through the non-linear system, the posterior mean and covariance are captured up to the third order of the Taylor series expansion of any non-linear system. Compared to the EKF that only captures dynamics from a first order Taylor expansion, the UKF has demonstrated a performance improvement in state estimation for non-linear systems \[14, 15\].

The UKF algorithm has a similar procedure as the standard Kalman filter, but with an extra step where the \(\sigma\)-points are calculated. The algorithm can be summarized as follows:

1. **Calculate \(\sigma\)-points and corresponding weights**

   \[
   \begin{align*}
   S_0 &= \hat{x}_{k|k} \text{ } \text{(2.21)} \\
   S_i &= \hat{x}_{k|k} + (\sqrt{(L+\lambda)P_{k|k}})_i \text{ } \text{(i = 1, 2, L)} \text{ } \text{(2.22)} \\
   S_i &= \hat{x}_{k|k} - (\sqrt{(L+\lambda)P_{k|k}})_{i-L} \text{ } \text{(i = L, L+1, 2L)} \text{ } \text{(2.23)} \\
   W_0^{(m)} &= \frac{\lambda}{L+\lambda} \text{ } \text{(2.24)} \\
   W_0^{(c)} &= \frac{\lambda}{L+\lambda} + (1-\alpha^2 + \beta) \text{ } \text{(2.25)} \\
   W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)} \text{ } \text{(i = 1, 2,..., 2L).} \text{ } \text{(2.26)}
   \end{align*}
   \]
2. Prediction Step:

\[ \hat{x}_{\sigma,i} = f(S_i, u_k, 0), \quad (i = 1, \ldots, 2L + 1) \]  
\[ \hat{x}_{k+1|k} = \sum_{i=0}^{2L} W_i^{(m)} \hat{x}_{\sigma,i} \]  
\[ P_{k+1|k} = \sum_{i=0}^{2L} W_i^{(c)} [\hat{x}_{\sigma,i} - \hat{x}_{k+1|k}] [\hat{x}_{\sigma,i} - x_{k+1|k}]^T + Q. \]  

(2.27)  
(2.28)  
(2.29)

3. Measurement Update Step:

\[ \hat{y}_{\sigma,i} = h(\hat{x}_{\sigma,i}, 0), \quad (i = 1, \ldots, 2L + 1) \]  
\[ \hat{y}_{k+1} = \sum_{i=0}^{2L} W_i^{(m)} \hat{y}_{\sigma,i} \]  
\[ P_{yy} = \sum_{i=0}^{2L} W_i^{(c)} [\hat{y}_{\sigma,i} - \hat{y}_{k+1}] [\hat{y}_{\sigma,i} - \hat{y}_{k+1}]^T + R \]  
\[ P_{xy} = \sum_{i=0}^{2L} W_i^{(c)} [\hat{x}_{\sigma,i} - \hat{x}_{k+1|k}] [\hat{y}_{\sigma,i} - \hat{y}_{k+1}]^T \]  
\[ K_k = P_{xy} P_{yy}^{-1} \]  
\[ \hat{x}_k = \hat{x}_{k+1|k} + K_k (z_{k+1} - \hat{y}_{k+1}) \]  
\[ P_{k+1|k+1} = P_{k+1|k} - K_k P_{yy} K_k^T. \]  

(2.30)  
(2.31)  
(2.32)  
(2.33)  
(2.34)  
(2.35)  
(2.36)

First the \( \sigma \)-points are calculated. Theses points are chosen such that they represent the true mean and covariance of the GRV by placing \( \sigma \)-points in the margin of the square root of the covariance matrix. The expression \( \sqrt{(L + \lambda) P_{k|k}} \) denotes the \( i \):th row of the matrix square root of \( (L + \lambda) P_{k|k} \) received by a Cholesky factorization. \( L \in \mathbb{R} \) is the number of states that should be estimated in the filter. \( \alpha, \beta, \gamma \in \mathbb{R} \) are tuning parameters of the filter that can be used to incorporate prior knowledge about the system.

In the prediction step, each of the \( \sigma \)-points is propagated through the non-linear system \( f \). To achieve an estimate of the predicted state, a weighted mean value of the propagated \( \sigma \)-points is calculated. With this weighted mean, the predicted covariance is estimated according to (2.29).

In a similar approach the measurements are predicted by a propagation of the sigma-points through the non-linear system \( h \). The cross covariance between the predicted state and the prediction of the measurement covariance are then estimated by (2.32) and (2.33). With these covariances, the state and covariance estimates can be obtained.
2.3.5 Filtering with errors-in-variables using UKF

The extended and unscented Kalman filters consider process and measurement noise. However, sometimes the model parameters in the prediction model and in the measurement model can also be uncertain and contain noise. In these cases, the covariance matrices uncertainty values can be augmented. However, uncertainties in model parameters cannot be mapped to an increase in the output uncertainty due to non-linear transformations. This means that the noise can affect the model output differently, depending on the current state.

One option is an errors-in-variables filtering technique that takes errors-in-variables into account. Such technique is known as the augmented unscented Kalman filter (AUKF). The AUKF extends the UKF state vector to contain the original state vector, an estimate of each process noise variable and an estimate of each measurement noise variable. With these estimates and their representation through the \( \sigma \)-points, noise in the model parameters could be incorporated in the model and also be more correctly accounted for in the model output. This augmented state vector can also be extended with estimates of model parameter noise. When the \( \sigma \)-points are calculated, the model parameter noise will be incorporated in each \( \sigma \)-point, which can be used in the unscented transform to represent the noise of model parameters.

It is worth noting that the AUKF will increase the number of states in the state vector, which also will increase the computational cost due to larger matrices and more \( \sigma \)-points. Since the number of \( \sigma \)-points is related to the number of states, \( N \), as \( 2N + 1 \), this means that two \( \sigma \)-points are added per each extra dimension in the state vector. To prevent the states from containing more state variables than needed, a combination of the UKF and the AUKF can be done, where the states that have independent additive noise to the outputs can be omitted and used as in the standard UKF instead [20].

The AUKF is presented in (2.37)- (2.54), where \( S_i = \begin{bmatrix} (S_i^x)^\top & (S_i^Q)^\top & (S_i^{QQ})^\top \end{bmatrix}^\top \).
1. Calculate \( \sigma \)-points and corresponding weights

\[
\tilde{x}_{k|k}^\sigma = \begin{bmatrix} \tilde{x}_{k|k}^T \ 0 \ 0 \end{bmatrix}^T
\]  
(2.37)

\[
P_{k|k}^\sigma = \begin{bmatrix} P_{k|k} & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}
\]  
(2.38)

\[ S_0 = \tilde{x}_{k|k}^\sigma \]  
(2.39)

\[ S_i = \tilde{x}_{k|k}^\sigma + (\sqrt{(L+\lambda) P_{k|k}^\sigma})_i \quad (i = 1, 2, \ldots, L) \]  
(2.40)

\[ S_i = \tilde{x}_{k|k}^\sigma - (\sqrt{(L+\lambda) P_{k|k}^\sigma})_{i-L} \quad (i = L, L+1, \ldots, 2L) \]  
(2.41)

\[ W_0^{(m)} = \frac{\lambda}{L+\lambda} \]  
(2.42)

\[ W_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \]  
(2.43)

\[ W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)} \quad (i = 1, 2, \ldots, 2L). \]  
(2.44)

2. Prediction Step:

\[ \hat{x}_{\sigma,i} = f(S_i^R, u_k, S_i^Q), \quad (i = 1, \ldots, 2L + 1) \]  
(2.45)

\[ \tilde{x}_{k+1|k} = \sum_{i=0}^{2L} W_i^{(m)} \hat{x}_{\sigma,i} \]  
(2.46)

\[ P_k^c = \sum_{i=0}^{2L} W_i^{(c)} [\hat{x}_{\sigma,i} - \tilde{x}_{k+1|k}] [\hat{x}_{\sigma,i} - \tilde{x}_{k+1|k}]^T. \]  
(2.47)

3. Measurement Update Step:

\[ \hat{y}_{\sigma,i} = h(\hat{x}_{\sigma,i}, S_i^R), \quad (i = 1, \ldots, 2L + 1) \]  
(2.48)

\[ \tilde{y}_{k+1} = \sum_{i=0}^{2L} W_i^{(m)} \hat{y}_{\sigma,i} \]  
(2.49)

\[ P_{yy} = \sum_{i=0}^{2L} W_i^{(c)} [\hat{y}_{\sigma,i} - \tilde{y}_{k+1}] [\hat{y}_{\sigma,i} - \tilde{y}_{k+1}]^T \]  
(2.50)

\[ P_{xy} = \sum_{i=0}^{2L} W_i^{(c)} [\hat{x}_{\sigma,i} - \tilde{x}_{k+1|k}] [\hat{y}_{\sigma,i} - \tilde{y}_{k+1}]^T \]  
(2.51)

\[ K_k = P_{xy} P_{yy}^{-1} \]  
(2.52)

\[ \hat{x}_{k+1|k+1} = x_{k+1|k} + K_k (z_{k+1} - \hat{y}_{k+1}) \]  
(2.53)

\[ P_{k+1|k+1} = P_{k+1|k} - K_k P_{yy} K_k^T. \]  
(2.54)
2.4 Outlier detection

In order to sort out which data are suitable for estimation and which data are not (probably coming from non-existing objects or non-stationary ones), an outlier detection layer is needed. This algorithm should label each of the sensor data points either as an outlier or as reliable data for the filtering algorithm. In what follows, different algorithms to achieve this task are discussed.

2.4.1 Definition

An outlier is a data point that significantly differs from the rest of the remaining data [21]. D. M. Hawkins [22] wrote that an intuitive definition of an outlier is:

"an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism".

In this thesis, two different kinds of data will be considered as outliers. The first type is data that differ significantly from the rest, to the extent that they can not be described by the underlying process model, e.g. a fault measurement. The other kind of data are points suspected to be derived by another process, e.g. a measurement from another moving object. Both of these types of events will be considered and refereed to as outliers.

2.4.2 Threshold-based outlier detection

With the use of linear correlation in a lower-dimensional sub-space of the data, a model can be estimated [21]. An assumption here is that the data have a high correlation in a lower-dimensional sub-space. This model can for instance be a hyperplane estimated using least-squares. Assuming that this model describes the data, an outlier can be detected by measuring the Mahalanobis distance [23] between the hyperplane and the data. The Mahalanobis distance measured can then quantify the outlier-score and the data can then be labeled by a thresholding policy. Here we provide three different methods that can account for outlier detection.

RANSAC

Random sample consensus (RANSAC) is one of the most successful techniques for robust data that may contain outliers [24]. RANSAC relies on constructing multiple model hypotheses based on a minimal sub set of data. Each model is then evaluated with the rest of the available data, and the validity of each model is then described by the number of data fitting the same model within some margin. A model using more data points can be associated with an improvement on its validity.

The minimal sub sets of data that are used to construct the models are randomly chosen from the data set and the validity of each model is described in its most simple form by counting the number of data points that are located within a threshold. The threshold of the model is related to the amount of noise that
is present in the data but the actual value can be designed using the statistical properties of the data. The procedure is described in Algorithm 1.

**Algorithm 1 Standard RANSAC**

1: procedure RANSAC(data, number_of_models, threshold)
2:    most_trusted_model = {}
3:    best_model_number = 0
4: for i = 1 to number_of_models do
5:    Sample a random minimal set of data → S
6:    measurement = data\S
7:    Estimate the model using S → T
8:    M = measurement.length
9: for j = 1 to M do
10:       Estimate measurement based on T → y(j)
11: end for
12: N_assoc = 0
13: for j = 1 to M do
14:       if ||measurement(j)-y(j)||<threshold then
15:          N_assoc = N_assoc + 1
16:       end if
17: end for
18: if N_assoc > best_model_number then
19:    most_trusted_model = T
20:    best_model_number = N_assoc
21: end if
22: end for
23: Return most_trusted_model
24: end procedure

Since RANSAC is based on choosing the data points randomly, the number of estimated models is an important input to the algorithm. With few models, all models might have been estimated using outliers. Thus, a reasonable amount of models needs to be considered. The number of hypotheses necessary, $N_{hypotheses}$, to ensure that at least one model is estimated without using outliers with probability $p$ is described in [24] as

$$N_{hypotheses} = \frac{\log(1-p)}{\log(1-(1-\epsilon)^m)}$$

(2.55)

where $\epsilon$ denotes the outlier ratio\(^1\) and $m$ is the minimum number of data points that are needed to instantiate the model.

The most trusted model and all data points that fit it can then be treated as valid data.

---

\(^1\) Number of outliers
----------------------
Total number of data
CHAPTER 2. BACKGROUND THEORY

Notice that RANSAC will require that the outliers do not come from a source that generates similar results for many outliers. If this is the case, outliers can generate the most trusted RANSAC model, and thus be treated as inliers. If the outliers generate the most trusted model, then it is more likely that the data that are correct data will be labeled as an outlier and the outlier data will be treated as inliers.

**χ²-testing**

χ²-testing can determine whether a sample of data matches a distribution. The test determines if the sample data point is an outlier or if it matches the population. The χ²-testing is built up by

\[
\chi^2_c = (z - \mu)\Sigma^{-1}(z - \mu)
\]

where \(z \in \mathbb{R}^d\) denotes a vector of data points, \(\mu \in \mathbb{R}^d\) is the expected value and \(\Sigma \in \mathbb{R}^{d\times d}\) is the standard deviation of the distribution. It can be shown that \(\chi^2_c\) is approximately \(\chi^2(n - 1)\) distributed \([25]\). This means that a higher value of \(\chi^2_c\) indicates that the data points does not fit well enough with the expected distribution. A lower value indicates that the data point matches the distribution better. The χ² test is then a binary classification of samples according to (2.56), where \(x^2_c > \text{threshold}\) classifies the sample as an outlier. The threshold can be chosen from a χ²-distribution, which is dependent on the degrees of freedom and the significance level.

The χ²-test determines if the data samples match a given distribution, but will not determine which data points do not match it. To use this as an outlier detection algorithm, the data are needed to be separated into several batches, or even into one test for each data point.

**Fraction update**

As mentioned in this section, RANSAC can, if there are many outliers that generate similar measurements, result in that the outliers are treated as valid data and that the correct measurements are treated as outliers. To prevent this, one option is to incorporate the previous state into the outlier detection that is similar to RANSAC. Regarding our problem, instead of choosing a minimal set of measurements and estimates of the velocity and the yaw rate with this set, a test update of the filter can be done. The result from the test update can then be used in the same evaluation process as in RANSAC, where the number of data points that can be associated with this test update is calculated. The test update is, just like RANSAC, an iterative process, where the update with most associated data will be used to determine which data are outliers based on a threshold policy.
2.5 Sensor theory

2.5.1 Doppler radar

The basic principle of radar sensors was discovered and tested in the late 1800s, but the interest and development of full-fledged technology radar did not occur until World War II [26]. Today, radar sensors are widely used in the society for applications such as weather analysis and assisted and autonomous driving. Radar sensors can be used both for positioning, tracking objects and to determine motions of objects relative to the sensor.

The radar itself is an electromagnetic sensor that operates by transmitting a particular type of wave-form and detecting the nature of the echo from the signal. From this signal, modern radar sensors can extract both a range measurement and a velocity measurement.

Range measurements

The range measurement is related to the time it takes for the signal to travel to the object and then be reflected back to the sensor. With a known electromagnetic propagation speed, \( c \), the range can be calculated with the relation between time, position and velocity. The relation is described in (2.57), where \( R \) is the one way radial measurement, \( T_{\text{transmission}} \) is the time when the signal is sent out and \( T_{\text{reflection}} \) corresponds to the time when the signal has returned back to the sensor.

\[
R = \frac{c(T_{\text{reflection}} - T_{\text{transmission}})}{2}.
\]  

Velocity measurements

The velocity measurement from radar data is based on a phase shift of the signal [27]. The phase shift \( \phi \) of a radar signal over a distance \( 2R \) is given by

\[
\phi = 2R \frac{2\pi}{\lambda},
\]  

where \( \lambda \) is the wave length. If the range \( R \) is changing linearly with time, there is a change in phase shift of the return echo [27]. This change depends on the relative velocity between the sensor and the object that reflected the signal and gives rise to a frequency change in the echo signal. This is known as the Doppler effect and the corresponding frequency changes is commonly referred to as the Doppler frequency [27].

To find the radial relative velocity, the first order derivative is calculated on each side of (2.58),

\[
\frac{d\phi(t)}{dt} = \frac{4\pi}{\lambda} \frac{dR(t)}{dt}.
\]  

The radial relative velocity, \( v_d \), is then identified as
\[ v_d(t) = \frac{dR(t)}{dt} = \frac{\lambda}{4\pi} \frac{d\phi(t)}{dt}. \]  \hspace{1cm} (2.60)

With the substitution \( \frac{d\phi}{dt} = w_d(t) = 2\pi f_d(t) \), where \( f_d(t) \) is the Doppler frequency, (2.60) becomes

\[ v_d(t) = \frac{\lambda f_d(t)}{2}. \] \hspace{1cm} (2.61)

### 2.5.2 Aliasing in Doppler radars

#### Range Folding

A pulse Doppler radar transmits a series of pulses that are separated by a distance, \( d \). The distance between two pulses is related to the pulse repetition frequency (PRF), through which the pulses are sent, and the speed \( c \) at which the pulse propagates [28]. The distance is

\[ d = \frac{c}{PRF}. \] \hspace{1cm} (2.62)

Thus, the maximum range, \( r_{\text{max}} \), that one pulse can travel and return to the sensor before the next pulse is transmitted is half of the separation distance \( d \),

\[ r_{\text{max}} = \frac{c}{2PRF}. \] \hspace{1cm} (2.63)

If a distance longer than \( r_{\text{max}} \) is measured, the next pulse will be sent out before the previous one has returned. The pulses are then folded into each other, which causes aliasing of the data [28]. The result of this is that the sensor cannot distinguish between the range \( r \) and \( r + nr_{\text{max}} \) for any integer \( n \geq 1 \). Thus, to avoid aliasing, there is a limit on the value of PRF.

#### Doppler velocity aliasing

The range aliasing would be possible to solve by just setting a low value of PRF. Then a long range measurement would be possible. However, this approach introduces difficulties for Doppler radar velocity measurements [28]. The maximum Doppler velocity measuring interval is related to the PRF and the radar wavelength, \( \lambda \), as

\[ V_{\text{max}} = \pm \frac{\lambda PRF}{4}. \] \hspace{1cm} (2.64)

Thus, the maximum possible velocity measuring interval decreases as PRF decreases. This introduces a trade-off between range measurements and velocity measurements. Substitution of PRF from (2.63) into (2.64) gives

\[ V_{\text{max}} r_{\text{max}} = \pm \frac{c\lambda}{8}. \] \hspace{1cm} (2.65)
The product of $V_{\text{max}}$ and $r_{\text{max}}$ becomes a constant under the assumption that the wave length, $\lambda$, is constant. This means that there is a trade-off between the range and velocity measurements.
Chapter 3

Method

3.1 Vehicle coordinate system

The vehicles used for collecting data and testing the system in real scenarios are Scania vehicles used for internal tests and development. In previous work, the local coordinate system of the vehicle has been set with its origin in the center of rotation. The x-axis is set to be orthogonal to the wheel axis with positive direction towards the front of the vehicle. The z-axis is set to be orthogonal to the ground and the y-axis is defined 90 degrees counter clockwise from the x-axis as depicted in Figure 3.1.

Although the center of rotation has been assumed to be fixed in the middle between the two rear-wheel boogie axles, it has been observed during this thesis work that the center of rotation is not fixed over time as the vehicle moves. Depending on the movement and turn rate of the vehicle, the center of rotation changes spatially with respect to the vehicle. This work also shows how critical this behavior is regarding velocity and yaw rate estimation, for which we have reserved a special discussion in Section 3.5.

3.2 Vehicle setup

As mentioned, the vehicles used for testing in this thesis are modified Scania trucks. The vehicles are equipped with multiple sensors that are used to drive autonomously. Along this thesis, only the Doppler radar sensors will be used to generate data, although more information could be retrieved from other kinds of sensors. There are five radar sensors mounted on the truck of three different types (different $r_{\text{max}}$): long-range, mid-range and short-range radars. There is one long-range radar and one mid-range radar mounted in the front of the vehicle and three short range radars mounted around the vehicle pointing in different directions. The three short range radars are placed as follows: two of the radar sensors are mounted in each front corner of the vehicle and the third is placed in the rear end. With this setup, the radar detections are spread around the vehicle to cover the surroundings. The
CHAPTER 3. METHOD

20

Figure 3.1. Vehicle coordinate system and coverage of long-range (green), mid-range (blue) and short-range (red) radars on the vehicle.

vehicle is also equipped with an Real Time Kinematic GPS (RTK-GPS), which will be assumed to be the ground truth for this thesis, in the sense that the measurements coming from it are supposed to provide the real position, velocity and yaw rate of the truck along time. An illustration of the radar sensor setup and the vehicle coordinate system is shown in Figure 3.1.

3.3 Models for the vehicle motion

The movement of the vehicle can be predicted using different models. A model that describes the vehicle movement more accurately can be more reliable, and thus provide better results. To use a parametric model, the parameters must be either known, or possible to estimate.

In this section, two different models with different complexity will be presented.

3.3.1 Constant velocity model

One of the simplest model for the vehicle is obtained by assuming constant velocity and yaw rate. Apparently, this model does not describe all the dynamics of the systems involved. But with a small time step, this assumption provides good predictions since the velocity and yaw rate are continuous functions that change
CHAPTER 3. METHOD

slowly in reality due to underlying physics of the vehicle. The Constant Turn Rate Velocity (CTRV) model can be described as

\[ \begin{align*}
    v_{t+1} &= v_t + r_t, \\
    \omega_{t+1} &= \omega_t + n_t,
\end{align*} \tag{3.1} \tag{3.2} \]

where \( r_t \) and \( n_t \) represent the process noise. These sequences are descriptions of how uncertain the model is. All the dynamics that the model does not take into account can be modeled as process noise. Thus, a simple model that does not handle dynamics in the model typically needs a higher process noise than a model that correctly takes more dynamics into account.

3.3.2 Constant acceleration model

In continuous time, the velocity at time \( t \) can be described as a summation of the velocity at a previous time \( s \) and the integral of the acceleration from time \( s \) to time \( t \) as

\[ \begin{align*}
    v(t) &= v(s) + \int_s^t a(\tau) d\tau + r(t), \\
    \omega(t) &= \omega(s) + \int_s^t \omega(\tau) d\tau + n(t). 
\end{align*} \tag{3.3} \]

If we assume that the acceleration is constant between sampling instances, we obtain the model

\[ \begin{align*}
    v_{k+1} &= v_k + a_k \Delta T + n^v_k, \\
    \omega_{k+1} &= \omega_k + \alpha_k \Delta T + n^\omega_k, \\
    a_{k+1} &= a_k + n^a_k, \\
    \alpha_{k+1} &= \alpha + n^\alpha_k,
\end{align*} \tag{3.4} \]

where \( \Delta T = t_{k+1} - t_k \) is the time between two samples. \( n^v_k, n^\omega_k, n^a_k \) and \( n^\alpha_k \) are terms for the noise of each process.

The CTRA-model takes more dynamics into account than the CTRV-model at the expense of higher model complexity.

3.4 Measurement model

In theory, the Doppler radar sensors mounted on the test vehicle can each deliver 64 different measurements in each time step. Each measurement data point consists of the relative radial Doppler speed, \( v_d \), and the corresponding azimuth angle, \( \theta_d \), of the measurement, as depicted in Figure 3.2.

The process of mapping the current state with the measurements can be explained in two steps. In the first step, the current state needs to be transformed
CHAPTER 3. METHOD

Figure 3.2. Measurements $v_d$ and $\theta_d$ from a sensor.

into velocities $v_x(t)$ in x-direction and $v_y(t)$ in y-direction, in the sensor position. The mapping from angular velocity to velocity in the point of interest is $\omega(t) r = v(t)$ where $r = \sqrt{x_s^2 + y_s^2}$ denotes the distance between the origin of rotation (described by the origin in the x-y plane) and the point of interest (described by $(x_s, y_s)$). In x-direction, the contribution of angular velocity is $\omega(t)y_s$. Thus, the velocity of the sensor in point $(x_s, y_s)$ is $v_{xsens}(t) = v(t) - \omega(t)y_s$. In y-direction the velocity is 0, but the angular velocity contribution is $\omega(t)x_s$, thus, $v_{ysens}(t) = \omega(t)x_s$. The velocity components are illustrated in Figure 3.3.

The translation from the velocity and yaw rate in the vehicle coordinate system to $v_x$ and $v_y$ in the sensor position $(x_s, y_s)$ can be described by the linear transformation

$$S = \begin{bmatrix} 1 & -y_s \\ 0 & x_s \end{bmatrix}. \quad (3.5)$$

The second step is a rotation. The contribution of $v_x(t)$ and $v_y(t)$ in the direction of the measurement will in this step be summed up into one velocity. The contribution of the velocity to the x-direction is $v(t) \cos \theta_d$ and in y-direction $v(t) \sin \theta_d$. By defining

$$M = \begin{bmatrix} \cos \theta_d & \sin \theta_d \end{bmatrix}, \quad (3.6)$$

the mapping from the current state vector $(v, \omega)$ to the predicted measurement $v_{d,\text{pred}}$ is the linear transformation

$$v_{d,\text{pred}} = M \ast S \ast \begin{bmatrix} v \\ \omega \end{bmatrix} = (v - \omega y_s) \cos \theta_d + \omega x_s \sin \theta_d. \quad (3.7)$$
The Doppler radar sensors are measuring the velocities the detected objects have in relation to the sensor. For stationary objects, these relative velocities have the same magnitude as the vehicle velocity, but in the opposite direction. Thus, to map the current state to the sensor measurements, a sign flip is needed:

\[ v_d = -v_{d,\text{pred}}. \]  

(3.8)

### 3.5 Center of rotation

The coordinate system used for the truck is set as described in Section 3.1. However, the assumption that the center of rotation is a fixed point in the truck might not be accurate enough. Prior to this thesis, it has been noticed that the center of rotation is moving between the two rear axes depending on the motion of the vehicle. For the measurement equations described in Section 3.4, this means that the matrix \( S \), described in (3.5), does not describe the mapping from the velocity and yaw rate in the vehicle coordinate system to the velocity in the sensor. In order to make this mapping more accurate, the exact center of rotation, \((x_c(t), y_c(t))\), is needed. Since all the model and measurement equations are derived at the center of rotation, the estimation can still take place even if it is moving. If the center of rotation is known and the estimation of velocity and yaw rate takes place in the center of rotation, the matrix \( S \) can be modified to

\[ S = \begin{bmatrix} 1 & -(y_s - y_c(t)) \\ 0 & (x_s - x_c(t)) \end{bmatrix}. \]  

(3.9)

In order to use this estimation of velocity and yaw rate in other applications, which assume that the velocity and yaw rate are estimated in the origin, the velocity...
needs to be calculated in the origin. The velocity in the origin is a sum of the velocity in the center of rotation and a contribution from the yaw rate. In mathematical expressions this can be written as

\[ v_o = v_c + y_c \omega_c, \]  

(3.10)

where \( v_o \) is the velocity in the origin and \( v_c \) and \( \omega_c \) is the velocity and yaw rate in the center of rotation.

The yaw rate is defined by the rotation around the center of rotation, thus, the yaw rate cannot be moved to be represented in the origin. In this thesis, the yaw rate will be assumed to be the same for the center of rotation and in the origin of the x-y-plane.

However, since the center of rotation is not known, it needs to be estimated. This task is carried out in Section 3.6.3.

3.6 Architecture of the estimation procedure

The proposed system in this master thesis can be divided into four different parts. The first part is a pre-processing step, where aliasing of the data is considered and early outlier detection can be done. The second step is the outlier detection step, where each of the data points will be classified either as valid data or as an outlier. The third step is the main estimation filter of velocity and yaw rate using a UKF. Finally, the fourth step is a post-processing step, where outputs from the UKF are smoothed out to get a smoother final result.

However, depending on the methods used for each of these four steps, the steps are not always fully separable. This results in that each of the four steps can be dependent on another step, which results in that several parts are needed to run in parallel.

3.6.1 Preprocessing

When the sensor data measurements are first delivered, the measurements can contain bias and gain. A simple model for this is that the measurement \( \tilde{z}_k \) is built up by a linear combination of the true measurement as

\[ \tilde{z}_k = G_p \hat{z}_k + b_p + e_k, \]  

(3.11)

where \( G_p \in \mathbb{R} \) is the gain and \( b_p \in \mathbb{R} \) is the bias for sensor \( p \), which generated the measurement \( \hat{z}_k \). \( e_k \in \mathbb{R} \) is a noise term from the measurement which is assumed to be Gaussian distributed. The bias and gain will be constant terms in every measurement from one sensor. With multiple sensors that have different bias and gains, the compensations of bias and gain become even more important, since the measurements from different sensors otherwise would conflict. To compensate for the bias and gain, the measurements \( \tilde{z}_k \) can be found as
where $\tilde{e}_k$ still is assumed to come from a zero mean white Gaussian process, but with another standard deviation compared to $e_k$.

With a bias and gain compensated measurement, the data still might be aliased as described in Section 2.5.2. Aliased data will deviate from the true velocity and will in later steps, without any preprocessing, be treated as outliers. However, aliased data are not necessarily outliers. Even if the aliased data are not in the correct range, the data can still contain information that is useful for the estimation.

In this section, a simple way to detect and fix aliasing is described.

Detecting data that are aliased is relatively easy for the human eye. The data will be shifted with a constant, $V_{\text{max}}$, compared to what the expected value is. An example of aliasing is shown in Figure 3.4. To detect those data points, the system needs to have an estimate of where the true value is located. This indicates that the preprocessing step operates with the output of the system as an input. However, since the preprocessing step is needed to produce the output, the output is not available in the preprocessing step. But since this estimation is an iterative process, the previous estimation is available. With sufficiently small time steps, an assumption that the velocity and yaw rate are constants can be made. From this assumption, an estimation of the velocity and yaw rate can be found.

These estimates can be propagated through the model equations to map the measurements. With an estimate of which region the measurements should be, possible aliased data and outliers can be filtered out for further processing.

For data points potentially suffering from aliasing, the aliasing factor $n^*$ can be found as $n^* = \arg \min \limits_{n \in \mathbb{N}} \hat{r}_k - n V_{\text{max}}$, where $\hat{r}_k$ denotes an outlier at time step $k$.

Thus, adding $n^*V_{\text{max}}$ to the measurement can create more valid measurements for the updating steps of the Kalman filter. It is important to mention that this step is a preprocessing step. Thus, the outcome of this step should be data that the filter can possibly use, but the actual use of the data will be decided in a later step. So even if some of the data points that become dealiased are outliers, this step will treat them as data that can be derived to the velocity and yaw rate. The algorithm is described in Algorithm 2.

3.6.2 Outlier detection

Deciding which measurements are actually generated by the process we want to model is a crucial step for the performance of the velocity and yaw rate estimation. If too few measurements are used, the filter will not have enough information to provide a good estimate, while if too many measurements are used, the estimate could be affected by another process. As mentioned in Section 2.4, outliers can either be data that cannot be mapped to any known process or data that can be mapped to some other process. The outliers mapped to some other process are more challenging to detect, since they typically are more similar to the process we
CHAPTER 3. METHOD

26

\[ \theta_d \neq 4 \neq 3 \neq 2 \neq 1 \]

\[ \phi_d \text{ (rad)} \]

\[ v_d \text{ (m/s)} \]

Figure 3.4. Red box containing aliased data points.

Algorithm 2 Fix Aliasing for a single data point

1: procedure DEALIAS DOPPLER DATA(measured_data, \( \hat{x}_{k+1|k} \), \( V_{\text{max}} \), \( C_1 \), \( C_2 \))
2: pred_data = mapFromStateToMeasurement(\( \hat{x}_{k+1|k} \))
3: residual = pred_data – measured_data
4: if \( || \text{residual} || > C_1 \) then
5: \( n_{\text{min}} = \arg\min_n || \text{residual} - n V_{\text{max}} || \)
6: if \( || \text{residual} - n_{\text{min}} V_{\text{max}} || < C_2 \) then
7: measured_data = measured_data + \( n_{\text{min}} V_{\text{max}} \)
8: end if
9: end if
10: end procedure

want to estimate. In the case of measuring the velocity and yaw rate of a vehicle using radar data, the outliers that cannot be mapped to some other process can be fault measurements from the radar. Measurements coming from another process will mainly come from moving objects in the surroundings. Unless the movement of the moving objects is known, the detections that come from these objects are impossible to incorporate into the velocity and yaw rate, since the mapping between the current vehicle movement and the relative velocity between the two objects is unknown. Thus, these detections need to be filtered out. Several different methods will be implemented and evaluated to recognize the best method for the proposed system. The methods are described in Section 2.4.

All the methods in this thesis are threshold-based approaches. The difference between the methods is the way of finding the values that should be used before using the threshold. This is a process of finding a good estimate of the current velocity and yaw rate of the vehicle with outliers in the dataset.
Finding the threshold value

Prior to this thesis, a correlation between the velocity and the outlier detection rate was detected. With a constant threshold, the outlier detection rate is higher when the vehicle has a higher velocity. Since the threshold is constant, more measurements are placed outside a region around the estimate of the current measurement, implying that the absolute difference between them is larger than the threshold. This implies that either the measurements are in a worse condition for higher velocities, or the prediction of the measurement corresponds less to the measurements in higher speed. If the measurements are in worse condition, i.e. contain more noise, the threshold can be increased to include more data points. This can be done by changing the constant threshold to a threshold that is a function of the velocity or by normalizing the threshold with the predicted velocity. However, this requires the Kalman filter to take this change into account and to adapt its parameter according to the threshold. On the other hand, if the prediction of the measurements does not match the measurements, the threshold should not be increased, instead the prediction model of the measurements should be considered to be dependent in the velocity. A third option to this phenomenon is that the outlier rate might do increase with higher velocities.

In order to set a threshold, the data points from different radars at different speeds can be analysed. If the vehicle is placed in a situation where there are just static objects around it, the outliers in the data can be sorted out manually for some time steps. Analysis of these data points can then show how the threshold should be set in order to perform outlier detection. However, it is important to keep in mind that non-static objects also need to be considered during outlier detection. Thus, data from similar situations, where outliers are present, also need to be considered to make sure the threshold that is small enough so that it does not include measurements from non-static objects.

3.6.3 Main filter

The main filter estimation in this thesis is based on a UKF. The filter will be a combination of the UKF and the AUKF in order to limit the number of states in the filter, which is related to the amount of computations needed in each time step.

State vector

The output from the system will be a vector containing the velocity in the origin of the vehicle in the direction of local x-coordinate system and the yaw rate measured at the center of rotation. However, to create a good estimation of the outputs, there are multiple other quantities that need to be estimated as well. To be able to use the model equations explained in Section 3.3.2, the linear and angular accelerations are needed. The center of rotation \((x_c, y_c)\) is also unknown and needs to be estimated. This results in a state vector, \(x_k\), with 6 states as
where $v_k$ is the velocity, $\omega_k$ is the yaw rate, $a_k$ and $\alpha_k$ represent the lateral and angular acceleration, respectively, and $(x_k^c, y_k^c)$ is the dynamic center of rotation.

To use the measurement models, $\sigma$-points for the noise in the measurement angles are needed to incorporate the noise of the measurement. With augmented states we get a state vector with 13 states (the noise for $v_d$ is omitted, since its noise is additive). The augmented state vector is described in (3.14):

$$
\begin{align*}
\mathbf{x}_k^a = \begin{pmatrix}
  v_k \\
  \omega_k \\
  a_k \\
  \alpha_k \\
  x_k^c \\
  y_k^c \\
  v_k^{\text{noise}} \\
  \omega_k^{\text{noise}} \\
  a_k^{\text{noise}} \\
  \alpha_k^{\text{noise}} \\
  x_k^{c,\text{noise}} \\
  y_k^{c,\text{noise}} \\
  \theta_d^{\text{noise}}
\end{pmatrix}.
\end{align*}
$$

Even if the state vector contains 13 different states, there are only 6 states that should be estimated and be propagated in time. The other states will be initialized to zero in each iteration and only matter in the propagation of the six states through model and measurement equations.

With this state vector in (3.13), the propagation of the state vector from $x_{k|k}$ to $x_{k+1|k}$ can be described as

$$
\begin{align*}
x_{k+1|k} = \mathbf{A} x_{k|k} = \begin{pmatrix}
  1 & 0 & \Delta T & 0 & 0 & 0 \\
  0 & 1 & 0 & \Delta T & 0 & 0 \\
  0 & 0 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} x_k,
\end{align*}
$$

where $\Delta T = t_{k+1} - t_k$ is the time between the two samples.
CHAPTER 3. METHOD

Measurement vector

Each radar sensor on the vehicle can in theory deliver 64 measurements in a time step. With \( m \) radar sensors, the maximum number of measurements that can arrive to the filter during a filter update is \( 64 \times m \) measurements. However, measurements are not delivered at the same time at a rate such that the sensors deliver \( 64 \times m \) measurements each time step. Thus, the filter will have a different amount of data to use at each time step. Each measurement contains a measurement of the radial relative speed between the vehicle and the current object, \( v_d \), and an azimuth angle, \( \theta_d \), that describes the direction in which the object is detected. The measurement equation to map the measurements to the predicted states \( v_k \mid k \neq 1 \) and \( w_k \mid k \neq 1 \) is described in Sections 3.4 and 3.5 and the final result from these sections is that for a measurement \( i \)

\[
v_{d,\text{pred}} = S_i M_i \begin{bmatrix} v_{k \mid k-1} \\ \omega_{k \mid k-1} \end{bmatrix} = v_{k \mid k-1} \cos \theta_d + (x_{s_k} - x_c) \sin \theta_d - (y_{s_k} - y_c) \cos \theta_d \omega_{k \mid k-1}.
\]

(3.16)

For one time step, we assume that \( n \) measurements are available. This means that the predicted measurement vector \( y_k \) will be

\[
y_k = -\begin{bmatrix} v_{d,\text{pred}1} \\ v_{d,\text{pred}2} \\ v_{d,\text{pred}3} \\ \vdots \\ v_{d,\text{pred}n} \end{bmatrix}
\]

(3.17)

and the measurement vector \( z_k \) from the radars is

\[
z_k = \begin{bmatrix} v_{d1} \\ v_{d2} \\ v_{d3} \\ \vdots \\ v_{dn} \end{bmatrix}.
\]

(3.18)

Filter initialization

The filter needs an initialization step for the first batch of data it is provided with, in order to derive a point to work around. The initialization gives the filter a good starting point, which is needed if the outlier detection is based on the current state. To make it possible to detect possible outliers of the data, an optimization problem with an \( \ell_1 \)-loss function is done as

\[
\min_{x_0} \sum_{i=1}^{n} |v_{d_i} - S_i M_i x_0|
\]

(3.19)
CHAPTER 3. METHOD

given that \((x_c, y_c) = (0, 0)\). This optimization problem is less sensitive to outliers than a least squares optimization problem \((\ell_2)\) since the errors of outlier with \(\ell_1\) become smaller in relation to the valid data than with \(\ell_2\). With the solution of the \(\ell_1\)-optimization, outliers can be found by using a threshold function such that if the absolute difference between the predicted measurement and the measurement is larger than a threshold, then it is an outlier.

Once outliers are sorted out, a least squares estimation can be done using the data that have been collected and treated as correct measurements. This least squares estimation of \(v\) and \(\omega\) can be used as initialization of the filter. All other states in the state vector are initially set to zero, since the filter can benefit from a good estimation of these states, but it is still not dependent on them for creating an estimate that can be used to update all other states.

3.6.4 Post processing

The UKF uses one measurement equation to update several states. In order to capture the dynamics in the states, the filter coefficients cannot be set to trust the model too much. This results in an estimate that is allowed to change fast and trust the measurements more. However, the estimates of some variables can still contain more noise than wanted for the purpose of the state. Such states are the lateral and angular accelerations. These states correspond to derivatives of the measured states, and numerical estimations of derivatives tend to be inaccurate and more sensitive to noise [29]. Thus, a post processing filter was implemented. The aim of this filter is to smooth the states that were used internally to better fit in the model. This filter was implemented with a model that assumed that the lateral and angular accelerations are constant between time steps. With this model, the filter can provide smoother estimates for the accelerations.

To make sure that the covariance matrix, \(P\), still is positive definite, which is a requirement for the filter to be able to run, this smoothing step needs to be done in cooperation with the unscented Kalman filter, where the Kalman gains that correspond to the accelerations are multiplied with the Kalman gain of the smoothing filter. This will ensure that the covariance matrix is positive definite.

3.7 Process and measurement covariances

The process noise covariance matrix \(Q\) and the measurement noise covariance matrix \(R\) represent the uncertainty of the model and the measurements. These parameters can be seen as parameters that should be tuned to gain accuracy of the UKF. Since these parameters can be chosen arbitrary, the process of tuning the filter is an optimization problem with many parameters. To find tuning parameters for the filter, several assumptions have been made:

- The noises of different measurements are mutually independent. Thus, the cross-correlation terms in \(R\) are all assumed to be zero. This means that the
measurement noise covariance matrix should be a diagonal matrix.

- The process noises for the velocity and the yaw rate are independent.
- The RTK GPS measures the true velocity and yaw rate of the vehicle.

With these assumptions, estimates of the variances of each of the parameters in the matrices $Q$ and $R$ can be done based on data. It is important to keep in mind that these estimates are just a guideline of which range each of the parameters’ optimal solution is. With these estimates as a start, further work with fine tuning can be done.

### 3.7.1 Doppler velocity measurement

To estimate the standard deviation of the Doppler velocity measurement, $v_d$, we assume that the angle measurement, $\theta_d$, is noise free. With the RTK GPS velocity, $v_{rtk}$, and yaw rate, $w_{rtk}$, a value for $v_{d,t}$ can be calculated as

$$v_{d,t} = (v_{rtk} - y_s w_{rtk}) \cos \theta_d + x_s w_{rtk} \sin \theta_d. \quad (3.20)$$

This calculated value can then be compared to the measured value $v_d$. The residual between the calculated value and the measured value is a representation of the noise from the measurement. When this procedure is done for multiple measurements, the statistical distribution of the residual can be estimated. The standard deviation of the statistical distribution can then be assumed to describe the standard deviation of the measurement noise.

### 3.7.2 Angle measurement

The estimation of the angle measurement, $\theta_d$, relies on similar assumptions as in Section 3.7.1. However, instead of assuming that the angle is noise free, now the assumption is that the Doppler velocity measurement, $v_d$, is noise free. Thus, an estimate of the true $\theta_{d,t}$ can be found by solving

$$v_d = a \cos \theta_{d,t} + b \sin \theta_{d,t}, \quad (3.21)$$

where $a = (v_{rtk} - y_s w_{rtk})$ and $b = x_s w_{rtk}$. This equation is possible to solve for $\theta_{d,t}$. Let $A = \sin \gamma = \frac{a}{\sqrt{a^2 + b^2}}$ and $B = \cos \gamma = \frac{b}{\sqrt{a^2 + b^2}}$. Then $A^2 + B^2 = 1$ and $\gamma$ has an unique solution in $[0, \pi]$:

$$v_d = a \cos \theta_{d,t} + b \sin \theta_{d,t}$$
$$= \sqrt{a^2 + b^2} A \cos \theta_{d,t} + \sqrt{a^2 + b^2} B \sin \theta_{d,t}$$
$$= \sqrt{a^2 + b^2} (\sin \gamma \cos \theta_{d,t} + \cos \gamma \sin \theta_{d,t})$$
$$= \sqrt{a^2 + b^2} \sin (\theta_{d,t} + \gamma).$$
Thus, the solution for $\theta_{d,t}$ is

$$\theta_{d,t} = \arcsin \frac{v_d}{\sqrt{a^2 + b^2}} - \text{sign}(\arcsin A) \arccos B$$

where

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

This calculated angle can then be compared to the measured angle and the residual can then give a standard deviation, as described in Section 3.7.1.

### 3.7.3 Model velocity and yaw rate noise

The process noise should describe how well the model predicts the next step. With a CTRV model, the process noise should describe how far away from the true next state the current state is. By using the RTK GPS measurements, a residual can be calculated as $x_t - x_{t-1}$. For the CTRA-model, an iterative process was used where the residual was calculated as $x_t - x_{t-1} - a\Delta T$. The acceleration came in this case from the filter estimate. This residual can then be used to calculate the process noise, using the procedure described in Section 3.7.1. It is however worth to note that this is not really an estimate of the process noise, just an initial guess to make the filter work. From this value, further tuning can be done.

### 3.7.4 Lateral and angular acceleration noise

For the lateral and angular accelerations, $a$ and $\alpha$, the RTK GPS used in the vehicles does not give an acceleration with accuracy that can be used as a reliable ground truth. However, the process noise for $a$ and $\alpha$ can be interpreted as how much the values of the accelerations can change between two time steps.

### 3.8 Tuning parameters of the unscented Kalman filter

With the UKF, three tuning parameters were introduced that are not used in the KF and in the EKF. These three parameters are $\alpha$, $\beta$ and $\lambda$. To explain the effect of each parameter and how they are connected, one more variable $\kappa$ is introduced. The parameter $\alpha$ determines the spread of the sigma points. With a larger value of $\alpha$, the sigma points will be spread in a larger area. The spread of the sigma points is crucial to capture the dynamics of the non-linear functions. With $\sigma$-points that are spread out too much, local dynamics around the state can be lost, but with a too small spread, the dynamics of the function is not taken into account at all.

The parameter $\beta$ is used to incorporate prior knowledge about the distribution. For Gaussian distributions, the optimal choice is $\beta = 2$ [15]. $\kappa$ is a secondary tuning
CHAPTER 3. METHOD

parameter that usually is set to 0. The value of \( \lambda \) is then chosen as \( \lambda = \alpha^2(L+\kappa) - L \) [15].

With the parameters of the UKF set, the process and measurement covariance matrices can be further tuned from their estimations described in Section 3.7.

To decide whether one set of parameters is better than another, two types of criteria can be set up. The first type is requirement of the system, a criterion that the system has to be able to handle, such as movement from other vehicles. If the requirements are met, the two parameter settings can be compared using tools described in Section 4.1. This is a trade off between robustness and performance.

3.9 Observability of the states

In order to be able to estimate states in the Kalman filter, the system needs to be observable [30]. A linear system described by

\[
\dot{x} = Ax + Bu \\
z = Cx
\]

(3.23)

(3.24)

is observable if and only if the observability matrix

\[
O = \begin{bmatrix}
C \\
CA \\
\vdots \\
CA^n
\end{bmatrix}
\]

(3.25)

has full rank, \( n \), where \( n \) denotes the number of states in the state vector [31].

For a non-linear system, a linearization of the system can be done to build up the observability matrix with a Taylor expansion where the second order and higher terms are neglected [30]. In this thesis, we have a system described by (3.15) and (3.17). The \( A \) matrix is already linear, but to find the \( C \) matrix, a linearization has to be done. The linearization is done by calculating the Jacobian of the measurement equation evaluated at the current state:

\[
\begin{bmatrix}
\frac{\partial v_d}{\partial y_{k|k-1}} \\
\frac{\partial w_d}{\partial y_{k|k-1}} \\
\frac{\partial w_r}{\partial y_{k|k-1}} \\
\frac{\partial w_i}{\partial y_{k|k-1}} \\
\frac{\partial \hat{x}_c}{\partial y_{k|k-1}} \\
\frac{\partial \hat{y}_c}{\partial y_{k|k-1}}
\end{bmatrix}
= \begin{bmatrix}
\cos \theta_d \\
(y_{c_i} - y_{s_i}) \cos \theta_{d_i} + (x_{s_i} - x_{c_i}) \sin \theta_{d_i} \\
0 \\
0 \\
-w_{k|k-1} \sin \theta_{d_i} \\
w_{k|k-1} \cos \theta_{d_i}
\end{bmatrix}
\]

(3.26)

The terms of the observation matrix are then
CHAPTER 3. METHOD

\[
(CA^n)^T = \begin{bmatrix}
\cos \theta_{d_i} \\
q_i \\
n \Delta T \cos \theta_{d_i} \\
n \Delta T q_i \\
-w_{k|k-1} \sin \theta_{d_i} \\
w_{k|k-1} \cos \theta_{d_i}
\end{bmatrix}
\]  

(3.27)

where \( q_i = (y_c - y_{s_i}) \cos \theta_{d_i} + (x_{s_i} - x_c) \sin \theta_{d_i} \).

The observation matrix for \( m \) different measurements is then

\[
O = \begin{bmatrix}
\cos \theta_{d_1} & q_1 & 0 & 0 & -w_{k|k-1} \sin \theta_{d_1} & w_{k|k-1} \cos \theta_{d_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cos \theta_{d_m} & q_m & 0 & 0 & -w_{k|k-1} \sin \theta_{d_m} & w_{k|k-1} \cos \theta_{d_m} \\
\cos \theta_{d_1} & \Delta T \cos \theta_{d_1} & \Delta T q_1 & -w_{k|k-1} \sin \theta_{d_1} & w_{k|k-1} \cos \theta_{d_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cos \theta_{d_m} & \Delta T \cos \theta_{d_m} & \Delta T q_m & -w_{k|k-1} \sin \theta_{d_m} & w_{k|k-1} \cos \theta_{d_m} \\
\cos \theta_{d_1} & 2 \Delta T \cos \theta_{d_1} & 2 \Delta T q_1 & -w_{k|k-1} \sin \theta_{d_1} & w_{k|k-1} \cos \theta_{d_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cos \theta_{d_m} & 2 \Delta T \cos \theta_{d_m} & 2 \Delta T q_m & -w_{k|k-1} \sin \theta_{d_m} & w_{k|k-1} \cos \theta_{d_m} \\
\cos \theta_{d_1} & 3 \Delta T \cos \theta_{d_1} & 3 \Delta T q_1 & -w_{k|k-1} \sin \theta_{d_1} & w_{k|k-1} \cos \theta_{d_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cos \theta_{d_m} & 3 \Delta T \cos \theta_{d_m} & 3 \Delta T q_m & -w_{k|k-1} \sin \theta_{d_m} & w_{k|k-1} \cos \theta_{d_m} \\
\cos \theta_{d_1} & 4 \Delta T \cos \theta_{d_1} & 4 \Delta T q_1 & -w_{k|k-1} \sin \theta_{d_1} & w_{k|k-1} \cos \theta_{d_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cos \theta_{d_m} & 4 \Delta T \cos \theta_{d_m} & 4 \Delta T q_m & -w_{k|k-1} \sin \theta_{d_m} & w_{k|k-1} \cos \theta_{d_m} \\
\cos \theta_{d_1} & 5 \Delta T \cos \theta_{d_1} & 5 \Delta T q_1 & -w_{k|k-1} \sin \theta_{d_1} & w_{k|k-1} \cos \theta_{d_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cos \theta_{d_m} & 5 \Delta T \cos \theta_{d_m} & 5 \Delta T q_m & -w_{k|k-1} \sin \theta_{d_m} & w_{k|k-1} \cos \theta_{d_m} \\
\cos \theta_{d_1} & 6 \Delta T \cos \theta_{d_1} & 6 \Delta T q_1 & -w_{k|k-1} \sin \theta_{d_1} & w_{k|k-1} \cos \theta_{d_1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cos \theta_{d_m} & 6 \Delta T \cos \theta_{d_m} & 6 \Delta T q_m & -w_{k|k-1} \sin \theta_{d_m} & w_{k|k-1} \cos \theta_{d_m}
\end{bmatrix}
\]  

(3.28)

To make sure that the system is observable, the matrix \( O \) needs to be full rank, which can be achieved for \( m \geq 3 \). Thus, to be able to run the state update part of the UKF correctly, at least three measurements from different azimuth angles are needed.
Chapter 4

Testing

This chapter describes the process of measuring the performance of the system. Both performance measurements and which driving scenarios that are important to evaluate will be discussed.

4.1 Performance assessment

This section describes the numerical indices that are considered in this thesis.

4.1.1 "Ground truth"

When possible, it is easier to measure the performance of the system if there is some ground truth signal of the estimate. The ground truth signal should show the correct answer to the estimation and be used as a reference that we want to aim for. Thus, in the best of worlds, the difference between the estimate and the ground truth should be as small as possible. In this thesis, the perfect ground truth is not available, since there is no existing method to get the exact speed and yaw rate of the vehicle. However, there are methods with available sensors that can estimate the speed and yaw rate up to a satisfying accuracy that can be interpreted as a ground truth, since these sensors will deliver better accuracy than the estimations evaluated in this thesis. This sensor is the RTK GPS, which will be interpreted as ground truth and reference signal.

4.1.2 Root-mean-square error (RMSE)

The root-mean-square error (RMSE) is a measure of how the estimation performs over time. The numerical value describes how one expects the method to perform in a general situation. Mathematically, the sampled RMSE is

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - x_{ref,i})^2}, \quad (4.1)
\]
where $x_i$ is the estimated value at time $i$, $x_{ref,i}$ is the true value at time $i$, and where $N$ is the number of samples have been estimated.

Although the RMSE is an index that describes the average performance for the system, it does not provide any information about how the system performs in different situations such as worst case scenarios. The RMSE does also require a sufficiently large amount of data to represent the behavior of the system well enough, since peak values in the difference between $x_i$ and $x_{ref,i}$ will be too dominant if just a few number of samples are available.

### 4.1.3 Maximum absolute error (MAE)

As a complement to RMSE, the maximum absolute error (MAE) can be used as a worst case scenario index. The MAE describes the difference between the true value and the estimated value in the worst situation. Thus, this index does not require a lot of data to represent the performance of the system, but it requires the worst case scenario to be present in the data. Formally, the MAE is defined as

$$\text{MAE} = \max_i |x_i - x_{ref,i}|.$$  \hspace{1cm} (4.2)

The MAE does not describe any typical performance of the estimation, in the sense that the estimation can be very close to the ground truth in all samples but one, where the MAE will indicate bad performance as soon as there is one large absolute-error peak. However, used together with RMSE, the MAE describes the system performance in a way that is comparable between different methods.

### 4.1.4 Autocorrelation

For discrete-time stochastic processes, the autocorrelation is an index describing the dependence between samples with a fixed lag. For a white process, the autocorrelation is 1 at $\tau = 0$ and zero otherwise, where $\tau$ is the lag. In system identification this is a powerful tool for evaluation when the considered stochastic process is a collection of residuals. The autocorrelation of the residual between the discrete time series $x_i - x_{ref,i}$ for all $i$ should be white noise if the model of the system describes the dynamics of the system in a satisfying way [32]. Thus, an autocorrelation plot (with respect to $\tau \in (\infty, \infty)$) can show whether the model used contains all parameters to describe the dynamic model or not. If there are some peaks in the autocorrelation function for $\tau \neq 0$, there are probably some dynamics in the model that are being omitted and should be considered.

The autocorrelation function can be also employed to verify the validity of assuming that the process and measurement noises are white.

### 4.1.5 Cross-covariance

The cross-covariance between two signals can be used to find time delays of the systems. The cross-correlation between two deterministic signals indicates how
much the two correlate at different lags. In mathematical terms

\[ \text{Corr}(x, y)(i) = \sum_{m=-\infty}^{\infty} x^*[m]y[m + i], \]  

(4.3)

where \( x \) is a scalar zero mean stochastic process and \( x^* \) is the complex conjugate of \( x \).

### 4.1.6 Histogram fit of Gaussian distributed data

A histogram plot based on data shows how common one quantity is in a data set. On the x-axis the values, or an interval of values are showed, and in the y-axis the number of times (or a normalized version of it) it appears in the data set. The histogram plot is a tool that visualizes the distribution of the data. If the assumption is that the data comes from a specific distribution and the parameters describing the distribution are estimated from the data, the histogram plot can be compared with the probability density function to gain evidence in that the assumption is valid.

### 4.1.7 P-P and Q-Q plot

In complement to the histogram plots, the Probability-Probability plot (P-P plot) and the Quantile-Quantile plot (Q-Q plot) can be used. The P-P and Q-Q plots visualize the empirical distribution function and the quantile function, by comparing the empirical distribution to a reference distribution [33]. The reference distribution may be a null hypothesis, so the P-P and Q-Q plots can be used to determine the goodness of fit. With a good fit, the plot should be similar to the line determined by \( x = y \). When the sample is close to the reference distribution, the points in the P-P plot lie close to the line \( x = y \) and when they differ, the points are located over or below the line. The Q-Q plot compares the quantiles of the distributions. The Q-Q plot can in many cases be S-shaped, which may indicate that one distribution is more skewed than the other or that one of the distributions has heavier tails than the other.

### 4.2 Overfitting and cross-validation

In this thesis, the development and evaluation of the system will be done using data from a truck at Scania CV AB. This means that tuning and evaluation will be based on real data. In order to evaluate the performance of a filter, validation data should be collected from different environments, velocities and weathers. These data can then be assumed to be representative to the typical drive, with which we can tune the parameters of the filter. However, since the Kalman filter provides several parameters that can be used to tune the filter to fit the data better, it is important to consider overfitting. Overfitting is a term that is related to the
difficulties of fitting a complex model to data. With a complex model, a lot of parameters can be chosen in such a way that the output of the filter fits the data. However, these parameters can also be used to fit noise into the model, which is highly undesirable. Thus, finding good parameters for the filter might not be equal to fitting the data to the model parameters as good as possible. This introduces a trade-off between tuning for having good performance on the data used for tuning and all other possible validation data.

In order to find this balance point, the performance of the filter should be evaluated on several different sets of data. One set can be used for training, and the others should be used for validation of the performance. This procedure is called cross-validation. When a validation set contains different scenarios (compared to the scenarios which the training data was based on), the performance could be evaluated without overfitting issues. To find a good reference set to use as validation set that is presentative enough, without being too large, is a challenge.

4.3 Test scenarios

In order to find test scenarios that reflect all possible situations the vehicle can face, different parts of the method can be considered. The first part of the system, the preprocessing, deals with aliasing and bias compensation of the sensors. To make sure that this part works in a correct manner, situations where aliasing takes place need to be considered. Aliasing mainly occurs in high speeds, thus, to make sure that the dealiasing works as expected, test data at high speeds are needed. The bias and gain compensation should be done on every measurement, and it is a requirement for the following parts. Thus, no extra testing is required for bias and gain compensation.

The next part of the system is the outlier detection. The outlier detection is needed in all different manoeuvres, independent of how the truck is moving. This makes the evaluation of this part even harder. To select sets of data that represent the performance of the outlier detection, a division into two types was done. One type focuses on acceptance, where the filter has to behave in a specific way and can not classify certain measurements as outliers. Such a test corresponds to a full acceleration or braking test. The other type is more focused on performance of the filter and how the outlier detection works during typical scenarios.

The main filter and the post processing will also require tests in all different situations. The tests from these parts are the ones that will indicate the performance of the whole system.

4.3.1 Collection of data for filter parameter tuning

The filter parameter tuning makes sure that the filter is able to estimate the velocity and yaw rate accurately. With badly tuned parameters the filter can diverge, resulting in a filter that does not associate any measurements.
CHAPTER 4. TESTING

While tuning, it is important to have as much data as possible to make sure that the method can handle all situations. But in order to make it possible to do within the scope of this thesis, the amount of training data needs to be limited, in order to limit the simulation time. Thus, there is a trade-off between the amount of data used and the time spent on tuning. Since this thesis mainly focuses on the algorithms, the tuning will be limited to three different sets of data. The first set of data is collected from a truck driving on the test track at Scania CV AB. The second and third sets of data are focused on acceptance tests of outliers. More specifically, the second set of data will be collected from a drive where the truck accelerates as fast as possible from 0 m/s to a predefined velocity and then performs a full brake back to 0 m/s again. The third set is data from where the vehicle will be standing still, but objects in its surroundings will move around it.

4.3.2 Scenarios to evaluate performance

To evaluate the performance of the system, several validation data sets were collected. The first data set focuses on moving objects in the surroundings, where the vehicle followed another vehicle driving at various speeds. The second data set was focused on an environment different to the tuning data sets. The tuning data sets were mainly from open areas, while this validation set was collected while driving around in an environment full of trees. The third data set is a short situation where the vehicle is driving up a steep hill.

These three data sets capture the main challenges identified in this thesis, and these data sets will be considered to measure the performance of the system.

4.4 Time delay compensation

In sensor fusion, the time delay introduced by different sensor needs to be considered to appropriately use the measurements for fusion of data from different sensors. These time delays also need to be considered by the different filters. Since the aim is to estimate the velocity and yaw rate in real time, a filter providing better estimates without incurring in time delays brings more value to a fusion phase than delivering time-delayed estimates. The speed of the filter is related to the algorithm used but also to the tuning parameters of the filter. This introduces a trade-off between real-time estimation accuracy and smoothness. The filter can, with low values of the measurements covariance matrix, $R$, be tuned to rely on that the measurements are more accurate than they actually are. This results in a filter that trusts the measurements too much and typically oscillates around the true states. However, in other applications that can use the velocity as an input, the velocity is needed to be smooth in order to get a satisfying output. Thus, the output needs to be in real time and be smooth.

The measure of the time delay $\Delta$ between two signals $x$ and $y$ can be formulated as an optimization problem.
\[ \Delta = \arg\min_{i} \sum_{k=-\infty}^{\infty} (x[k+i] - y[k])^2. \quad (4.4) \]

While comparing the different algorithms, the time delay introduced by the sensors will be considered, but the time delay introduced by the filter will be treated as a part of the result. The time delay introduced by the filter is neglected since a more smooth tuning typically would claim a better result with time delay compensation, making it a fair assumption.
Chapter 5

Results

In this chapter, the performance of the proposed system is presented. The test scenarios are the ones described in Section 4.3. In order to isolate each part of the system, different versions of the system will be used to evaluate the proposed estimation method. All velocities are presented in m/s and all yaw rates are presented in rad/s.

5.1 Ground truth prefiltering

The ground truth signal in this thesis, the raw data of the RTK GPS, is considered to contain noise in the measurements, especially in the yaw rate. In order to get a more smooth signal, a standard Kalman filter is used with a CTRV-model. The result of the preprocessing filter is shown in Figure 5.1 and in more detail in Figure 5.2. It can be seen in Figure 5.2 that the yaw rate is smoother than the raw signal.

![Figure 5.1. RTK GPS: Raw signal and filtered signal on tuning data.](image-url)
5.2 Filters implemented to compare with the proposed system

In order to isolate each part of the system, different variations of the system will be used to evaluate the proposed system. For evaluation of the prediction model, the CTRA-model is compared to a system that is identical, but with less states, and a CTRV-model. Regarding the center of rotation estimation problem, a system with and without taking the dynamics of the center of rotation into account is to be compared. The performance of the UKF is compared to an implementation of the EKF and the overall performance is compared to the one of the methods currently used in the vehicles today. An overview of the systems is shown in Table 5.1.
5.3 Tuning parameters

5.3.1 Estimation of the measurement noise

For parameter tuning, data sets described in Section 4.3.1 are used. With a data set collected during 180 seconds, the noise terms was estimated according to Section 3.7. The histogram plots with a Gaussian distribution fit are shown in Figure 5.3. These histograms show that the assumption of zero mean Gaussian noise seems to hold, but with a bit heavy tails, as can be seen both in the histograms in Figure 5.3 and in the Q-Q-plots in Figure 5.4. The estimation process for a single sensor is shown in Figure 5.5. It can be seen that the noise distribution is sensor dependent.

![Figure 5.3](image)

Figure 5.3. Measurement noise estimation for $v_d$ and $\theta_d$. The blue histogram shows the normalized distribution of the residuals. The red lines shows the corresponding best Gaussian fit.

5.3.2 Estimation of process noise

With the parameters for the process noise estimated according to Section 4.3.1, the histogram plots and a corresponding Gaussian distribution are plotted in Figure 5.6. From this, the standard deviations for the processes can be found. Just as in the case of the measurement noise, the assumption that the process noise is Gaussian distributed seems to be valid. Yet, as seen in Figure 5.7, the autocorrelation for the velocity with CTRV-model does not verify the assumption that the noise comes from a white noise process.

5.3.3 Outlier threshold

For different velocities, the standard deviations of the measurements were analyzed. In Figure 5.8, the residuals between each measurement in the tuning set and the
corresponding predicted measurements (based on RTK GPS) are shown. It can be seen that there is a linear component as a function of the velocity in the residuals, which is indicated by the red lines. Thus, these lines could be used as a threshold policy for the outlier detection.

However, the threshold that was based on data from a typical drive does not represent the worst case scenarios. Thus, the threshold was required to have a larger constant offset in order to be able to handle situations where the vehicle accelerates or brakes as much as possible. This was set by trial and error based on the results from Figure 5.8. The slope of the velocity dependent threshold was determined by Figure 5.8 and the constant term was increased until all situations could be handled. In Figure 5.9, the behavior of the system on a full acceleration and brake test data set is shown before and after the final tuning.

5.4 Model evaluation

This section will evaluate the CTRA-model and compare it to the CTRV-model. The evaluation is related to the process noise estimation in Section 5.3.2. In Figure 5.6, the process noises were estimated. These process noises can be interpreted as how accurate the model is and how far away from the true value the predictions are. In Figure 5.6 it can be seen that the standard deviation of the best Gaussian fit is a little bit smaller for the CTRA-model than for the CTRV-model for the velocity and very similar for the yaw rate. In Figure 5.7, the autocorrelation function of the residuals shows that the CTRA-model residual is more likely to be white.

To measure the model performance in the filter, a study of the innovations shows
CHAPTER 5. RESULTS

Figure 5.5. Noise estimation with measurements from a single sensor. The plots show the estimation from the three types of sensors mounted on the vehicle: Long Range, Mid Range and Short Range Radar.

whether the predictions are more accurate in the CTRA-model than in the CTRV-model. In Figure 5.10, a plot of the RMSE of the point-wise squared innovation over time is shown. From this plot, it can be seen that the innovations using an UKF with CTRA-model are smaller than for an UKF with CTRV-model.

In Table 5.2, the RMSE and the MAE are shown for the UKF with CTRA- and CTRV-model on a ~190 seconds (18812 samples) long validation data set driven in different velocities and yaw rates. The bars are normalized so that the UKF with the CTRV-model is the reference (100 %). Thus, a bar with a lower value than 100 % indicates that the system performs better in terms of that measurement. As seen in Table 5.2, all values are below 100 %. Thus, the CTRA-model performs better
than the CTRV-model in these performance indices.

<table>
<thead>
<tr>
<th></th>
<th>RMSE $v$</th>
<th>RMSE $\omega$</th>
<th>MAE $v$</th>
<th>MAE $\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKF$_v$</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>UKF$_\omega$</td>
<td>83.28%</td>
<td>68.71%</td>
<td>67.12%</td>
<td>80.42%</td>
</tr>
</tbody>
</table>

Table 5.2. RMSE and MAE for the unscented Kalman filter with CTRV- and CTRA-model.
5.5 Outlier detection performance

The evaluation of the outlier detection is done using the UKF with CTRA-model, with the tuning parameters found in Section 5.3. The outlier detection will be evaluated in two different scenarios: one where the vehicle drives in different velocities without surrounding vehicles, and one scenario where the vehicle is surrounded by other moving objects. In Table 5.3, the RMSE and MAE are presented for the drive without many moving objects in the surroundings. This table shows that the outlier detection with a linear threshold around the predicted state has the best performance among the three methods. Regarding the yaw rate, the three methods have a quite similar performance.
Figure 5.8. Residuals as a function of the velocity. Each blue ball indicates the residual for a measurement. The red lines are used to indicate that there seems to be some dependence between the magnitude of the residuals and the velocity.

Figure 5.9. Behavior of the system before (left) and after (right) the final tuning of the outlier threshold.

A plot of the velocity and the yaw rate with linear-threshold outlier detection, when the vehicle is driving behind another vehicle in matching speed (thus all measurements from the front radar should be detected as outliers), is shown in Figure 5.11. It can be seen that the system can estimate the velocity and yaw rate, even if all measurements from the front sensors are outliers.
CHAPTER 5. RESULTS

Figure 5.10. Innovations over time for the unscented Kalman filter with CTRV- and CTRA-model.

<table>
<thead>
<tr>
<th></th>
<th>RMSE $v$</th>
<th>RMSE $\omega$</th>
<th>MAE $v$</th>
<th>MAE $\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKF$_a$</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
</tr>
<tr>
<td>UKF$_r$</td>
<td>101.59 %</td>
<td>102.18 %</td>
<td>103.92 %</td>
<td>100.88 %</td>
</tr>
<tr>
<td>UKF$_p$</td>
<td>103.34 %</td>
<td>99.67 %</td>
<td>105.57 %</td>
<td>100.61 %</td>
</tr>
</tbody>
</table>

Table 5.3. RMSE and MAE for the unscented Kalman filter with three different outlier detection algorithms without moving objects in the surroundings. UKF$_a$ is the UKF with a threshold on the innovations, UKF$_r$ uses RANSAC as outlier detection and UKF$_p$ uses the partial update outlier detection described in Section 2.4.2.

5.6 Center of rotation

The estimation of the center of rotation is shown in Figure 5.12. The center of rotation was estimated in both the x-axis and in the y-axis. The x-axis was initialized to zero, but seems to be constantly placed more towards the front rear axis. In the y-direction, the center of rotation has a smaller magnitude than in the x-axis and overall it seems to be placed a little bit more to the right than expected. The difference in performance of the system is shown in Table 5.4. When the dynamics of the center of rotation are considered, it can be seen that both the RMSE and the MAE are better for the yaw rate estimation, while the RMSE is better for the velocity. The maximum absolute error is very similar for the velocity.
CHAPTER 5. RESULTS

5.7 Unscented Kalman filter performance

In Table 5.5, the performance of the UKF is compared to the EKF, where the EKF implementation assumes that the angle measurement, $\theta_d$, is noise free. Perhaps surprisingly, both the filters achieve similar accuracy for both RMSE and MAE, and this phenomenon will be discussed in the following chapter.

Figure 5.11. Estimation when driving behind another vehicle. All measurements from the front radars are outliers.

Figure 5.12. Estimates of the center of rotation over time.
CHAPTER 5. RESULTS

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE $v$</th>
<th>RMSE $\omega$</th>
<th>MAE $v$</th>
<th>MAE $\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKF$_a$</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
</tr>
<tr>
<td>UKF$_{ac}$</td>
<td>93.13 %</td>
<td>79.47 %</td>
<td>99.99 %</td>
<td>78.44 %</td>
</tr>
</tbody>
</table>

Table 5.4. RMSE and MAE for the UKF$_{ac}$ compared and scaled by the RMSE and MAE from UKF$_a$.

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE $v$</th>
<th>RMSE $\omega$</th>
<th>MAE $v$</th>
<th>MAE $\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKF$_{ac}$</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
</tr>
<tr>
<td>EKF$_{ac}$</td>
<td>104.30 %</td>
<td>99.21 %</td>
<td>100.38 %</td>
<td>100.04 %</td>
</tr>
</tbody>
</table>

Table 5.5. RMSE and MAE for the UKF and the EKF.

5.8 Overall performance

The performance in the estimation of the velocity and the yaw rate is shown in Figure 5.14 and Figure 5.15, both with and without estimation of the center of rotation. Here, the filtering algorithm performance is compared to the current implementation. The RMSE and MAE of the same data set is shown in Figure 5.13. Comparison between the different methods shows that the proposed filtering algorithm performs better than the current method in all performance measures.

The estimated lateral acceleration is shown in Figure 5.16. The ground truth has been estimated with a numerical derivative of velocity from the RTK GPS and filtered by a median filter with 20 time steps. It can be seen that the predictions follow the patterns of acceleration, even if the latter contains different levels of noise.
Figure 5.13. RMSE and MAE for the UKF compared and scaled by the RMSE and MAE of the current version of the implementation based on least squares.
CHAPTER 5. RESULTS

Figure 5.14. Estimates of the velocity and yaw rate compared to the RTK GPS.

Figure 5.15. Detailed plot of estimates of yaw rate compared to the RTK GPS.
Figure 5.16. Estimate of acceleration compared to the computed acceleration based on RTK GPS data.
Chapter 6

Discussion

This section presents an analysis of the performance and characteristics of the proposed system.

6.1 Performance of the filtering algorithm

The performance of the proposed filtering algorithm seems promising. Compared to the current implementation, the proposed systems seems to perform better or equally good in all tests. With the proposed method, both the velocity and the yaw rate are able to follow the true values, but the measurements seem to describe the dynamics of the velocity better than the ones of the yaw rate. In fact, this can be seen in Figure 5.11, where the estimation of the yaw rate follows the dynamics of the true values well for smaller magnitudes, but does not follow the reference with the same accuracy for higher ones. In Figure 5.14, similar results are shown, but it also compares the system with and without the center of rotation estimation. From this figure and Figure 5.15, we can identify that the incorporation of the center of rotation seems to improve the estimation for higher magnitudes of yaw rate. These results can be motivated by looking at the measurement equation in (3.16), where an error in the sensor position affects the terms considering yaw rate, but does not affect the terms considering velocity.

The plots of the lateral acceleration shown in Figure 5.16 indicate that the estimate of acceleration works, even if the the acceleration may not follow the considered true value accurately at all time. However, a very high accuracy of the acceleration is not needed to make the model prediction better, because as long as the accelerations are fairly correct, the prediction phase can benefit from an acceleration estimate. It is important to keep in mind that the accelerations are not considered as an output of the system. The accelerations are estimated to be able to create a better prediction of the state. Thus, based on the RMSE and MAE shown in Table 5.2, the accelerations are shown to improve the state estimates.
CHAPTER 6. DISCUSSION

6.2 Tuning

The different measurements of the tuning parameters in Section 5.3 show that several of the assumptions are valid. In Figure 5.6 and Figure 5.7, it can be seen that the process noise for the CTRA-model seems to be generated from a zero mean Gaussian white process. For the velocity, it can also be concluded that the standard deviation of the Gaussian fit is slightly smaller for the CTRA-model compared to the CTRV-model. It can also be seen that the autocorrelation of the velocity with the CTRV-model does not verify the white assumption to the same level as the CTRA-model does.

The yaw rate process noise estimation is very similar when the CTRA- and CTRV-models are compared. A reason for this could be that the data set that was used mainly focused on driving in different speeds. The speed for which the most of the data were collected from was between \(5 \leq v \leq 20\) m/s. To get a comfortable ride, the yaw rate is typically low for high velocities. This indicates that the tuning set generated small values of angular acceleration, \(\alpha\), which results in a similar performance for both the CTRA-model and the CTRV-model.

The histogram plot in Figure 5.3 shows the residuals for all sensors mounted on the vehicle. This resulted in a histogram whose tail is heavier than the one of a Gaussian distribution, as seen in the Q-Q-plots in Figure 5.4. This is a result of that different types of sensors are used. In Figure 5.5, histograms for each sensor type that is mounted on the vehicle are shown. These histograms show that the noise level in \(v_d\) is different, depending on if the sensor is long-, mid- or short-range. \(v_d\) seems to be noisier for short-range sensors. Since there are multiple short-range sensors mounted on the vehicle, these measurements introduce a higher noise variance. The user then has to decide between having an accurate model for noises and the amount of tuning parameters of the filter. If each sensor would have an individual tuning, the number of tuning parameters to estimate would increase in two parameters per sensor instead of having two parameters for all sensors. Introducing multiple parameters increases the risk of overfitting and the possibility that the filter will diverge. Thus, there is a trade-off between the number of parameters and the amount of time that is needed for tuning. In this thesis, the measurement noises were set to a common value for all sensors, due to limited time spent on tuning.

It should also be mentioned that even if the estimation of the standard deviations of the noises in Figure 5.3 intuitively seems to be in the correct range, these estimations also contain noises that are derived from other sensor measurements, and then it may differ from the real value.

6.3 Prediction and measurement models

The CTRA-model shows good improvements compared to the CTRV-model in all measurements presented in Section 5.4. The fact that the innovations in Figure 5.10 became constantly smaller indicates that the prediction model moves the states
closer to the measurements. This is a good indication of that the CTRA-model works better than the CTRV-model. In order to use the CTRA-model an estimation of the angular and lateral accelerations was needed. This increases the number of states in the state vector, but these two states improve the performance of the filter, as seen in Table 5.2.

The measurement model has turned out to be a more interesting topic to discuss. Even if the mapping from the ego-motion to the radar measurements is simple, there are several parameters that are needed in order to be able to perform the mapping. These parameters can be measured or estimated, but measurements and estimations will not guarantee that the parameter is set in a correct way. The sensor position, the biases and gains that were estimated prior to this thesis have some uncertainty leading to an error if they were chosen incorrectly. Since the biases and gains affect the measurements and the sensor position affects the measurement model, these parameters need to be estimated using additional sensors as input.

In Figure 5.9, it is shown how the outlier detection performed before and after the final tuning of the outlier detection parameters. We remark that a model with constant acceleration is beneficial if the filter diverges at some time. It is clear that all radar detections are classified as outliers at the time at which the estimation diverges. Thus, the filter relies on the model, which makes a prediction for each time step. In this case, with a CTRA-model, it means that the prediction is that the velocity is a bit lower for the next state compared to the current state. This is an assumption that makes it possible for the filter to generate good estimates faster than with a CTRV-model. With a CTRV-model, the prediction would be a straight line when the filter does not associate any measurements. Thus, in order to create a good estimate of the states again, the vehicle needs to come back to a similar speed as when the last data were associated.

6.4 Outlier detection

The outlier detection testing indicates that the outlier detection works well. All algorithms seem to be able to identify outliers such that the outliers does not make the filter diverge, which is the most important role of the outlier detection. Thus, the tuning should be set to be conservative, to make sure that the filter does not diverge.

The three algorithms that were tested in this thesis seems to be similar in performance. Depending on which data set was used, the different algorithms performed more or less equally good. Thus, for the proposed system, the simplest outlier detection was chosen: threshold based outlier detection on the innovations of the Kalman filter.

A more interesting issue is the choice of threshold value. It has been observed during the thesis that the choice of threshold value strongly influences the performance of the filter. The threshold policy can be the difference between performing better or worse than the current system. This shows that the outlier detection is a
very important part of the system.

The outlier detection is not limited to estimation of velocity and yaw rate, but can also be used in object tracking and environmental analysis. Since measurements coming from non-static objects typically arrive in a group, it could be beneficial to look into outlier detection algorithms that cluster the detections into objects and keep track of them during multiple time steps. Creating a prediction of how these objects move will also make it easier to detect outliers. This will not only make outliers coming from static objects easier to detect, but also easy to detect outliers coming from non-stationary objects. Such a scenario is shown in Figure 6.1. Here, the vehicle has a moving object, in this case another vehicle, in positive $x$- and $y$-direction that generates detections from a moving object. A radar measurement in $y$-direction ($\theta_d = 90^\circ$) would be the same, independent of whether the object is static or moving in $x$-direction. But detections that deviate more from $\theta_d = 90^\circ$ would also claim more and more different velocities, depending on whether the object is static or moving. Thus, with a threshold based outlier detection, the measurements close to $\theta_d = 90^\circ$ would be very similar for both static and moving objects. However, if the measurements would have been clustered, the information from measurements from a moving object far away from $\theta_d = 90^\circ$ could be used to exclude detections where $\theta_d$ is close to $90^\circ$.

### 6.5 Center of rotation

The dependence of the estimation performance on the center of rotation was detected during this project. This is an issue that was not identified during the literature study, since most of it come from studies on cars. The dynamics of the center of rotation are mainly introduced by the fact that there is more than one rear axis on the vehicle. On a vehicle with one rear axis, e.g. a car, the center of rotation
will probably not move to the same extent as with multiple rear axes. Thus, if the vehicle has one rear axis, the assumption that the center of rotation is static at the center of that rear axis seems valid, as [2] assumes. But with more than one rear axis, the assumption that the center of rotation is placed in the middle between the two rear axes is not good any longer. The fact that the center of rotation moves is depicted in Figure 5.12, where the center of rotation moves depending on how the vehicle moves. However, since a ground truth for the center of rotation is not available, the performance of the estimations is hard to evaluate and tune. The tuning parameters for the center of rotation were tuned by hand and evaluation can just be done by quantitatively. An initial guess of where the center of rotation would be when the yaw rate is high in magnitude is that the center of rotation is placed on the front rear axes and that the other rear axis is slipping more than the front rear axis. The length between the two rear axes can be measured in order to evaluate if the estimate seems to be correct. If this is the case, then the center of rotation should be placed close to the front rear axis at high magnitudes of yaw rate. The center of rotation can also be hard to estimate due to the fact that the sensor position might not be very accurate. If the sensor position is not accurate, the bias in the sensor position will be incorporated in the estimation of center of rotation. With measurements from multiple sensors, an uncertainty in the sensor positions means that the measurements can conflict and indicate that the center of rotation should be placed at different places. Thus, in order to get a good estimation, it might be a better idea to estimate the sensor position in relation to the center of rotation. This results in that conflicting measurements update different variables, instead of the same variable. This would however increase the number of states in the filter, which also would affect the computational complexity of the system.

6.6 Estimation method

The main reason for using the UKF instead of an EKF in this thesis is that the UKF could perform non-linear transformations and handle model uncertainties better. However, as seen in Table 5.5, the performances of the UKF and the EKF are very similar. This indicates that the UT and a linearization seem to perform equally well in this case. One reason for this could be that the models do not have any dynamics where a linearization is far behind the UT when it comes to accuracy. Instead, changing the algorithms to update the state directly with each measurement and changing the prediction model seem to be the parts that have gained more accuracy.
Chapter 7

Conclusion

This chapter summarizes the work done in this thesis and proposes some areas for future work.

7.1 Thesis summary

This thesis project has investigated and presented a filtering algorithm for estimating the lateral velocity and yaw rate of a vehicle based on Doppler radar measurements. The work has been tested and evaluated on trucks equipped with multiple Doppler radar sensors at Scania CV AB. The proposed algorithm estimates the velocity and yaw rate of the vehicle by means of an unscented Kalman filter, once aliased data have been considered and outliers have been dropped out. Compared to the current implementation, the proposed filtering algorithm shows an improvement in RMSE by 28.8% for the velocity and by 22.4% for the yaw rate. Including an estimate of the lateral and angular accelerations and the center of rotation has shown improvements in the prediction models. Tests performed on real time data show that the proposed system performs better than the current system. However, the UKF seems to perform equally good as the EKF, which in many cases can be easier to tune and would consume less resources on the on-board processor.

7.2 Future work

Using a UKF to estimate the velocity and yaw rate based on radar detections has been proven be an attractive alternative to the least squares estimation method. One of the most obvious and most natural extensions to this work is more fine-tuning of parameters and more extensive tests. This should be done to find an even more robust and even better performing system.

In regards of the data association, the outlier detection has turned out to be very challenging, and also crucial for the performance of the algorithm. As mentioned in Section 6.4, a future question to answer is whether keeping track of objects can improve the outlier detection. Dynamical objects are typically present in detections
during multiple time steps, which means that information about detections from previous time steps could improve outlier detection of the current time step.

The center of rotation, which seems to move dynamically depending on the movement of the vehicle is also an interesting research question for future work. As mentioned in the discussion, estimations of sensor positions with respect to the center of rotation could be one option for improving the estimation.
Bibliography


