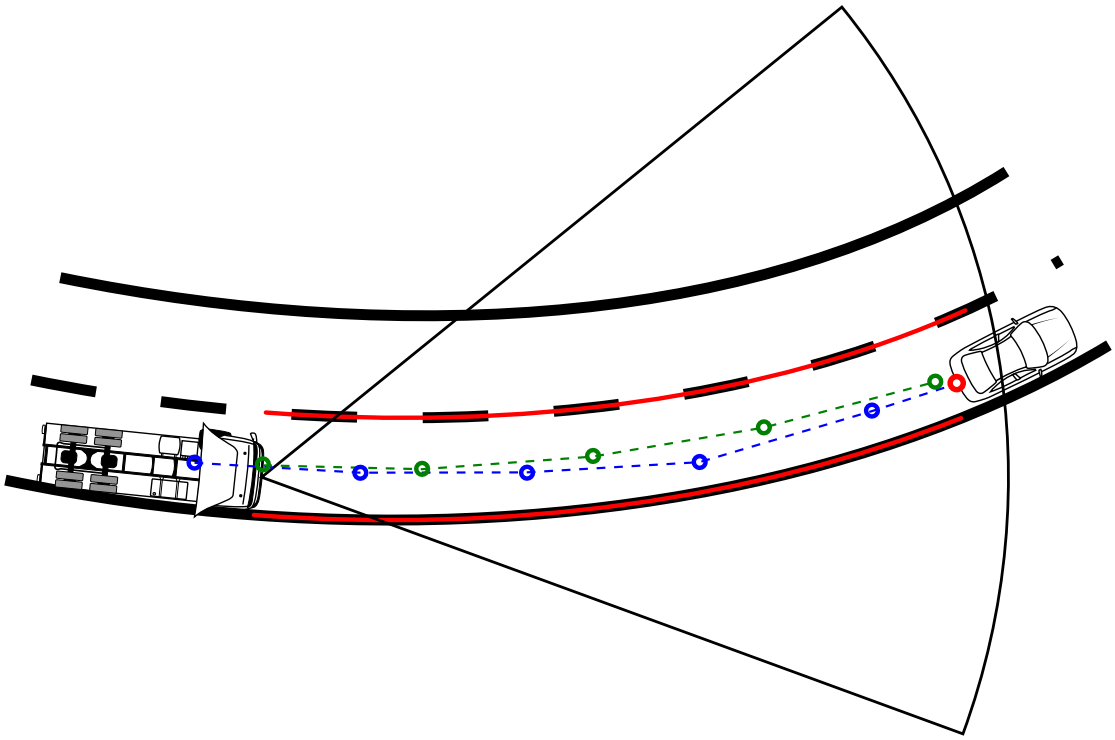
CHALMERS
UNIVERSITY OF TECHNOLOGY



Reference Path Estimation for Lateral Vehicle Control

Master's thesis in Systems, Control and Mechatronics

Arni Thorvaldsson

Vinzenz Bandi

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Master's Thesis Ex041/2015

Reference Path Estimation for Lateral Vehicle Control

Arni Thorvaldsson
Vinzenz Bandi

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden 2015

Reference Path Estimation for Lateral Vehicle Control
Arni Thorvaldsson
Vinzenz Bandi

Supervisor: Erik Nordin & Mansour Keshavarz, Volvo Group Trucks Technology
Examiner: Lennart Svensson, Signals and Systems

Master's Thesis 2015
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2015

Reference Path Estimation for Lateral Vehicle Control
Arni Thorvaldsson
Vinzenz Bandi
Department of Signals and Systems
Chalmers University of Technology

Abstract

Autonomous driving cars have been a hot topic in the media in recent years, with more and more tech companies and universities presenting projects with fully automated vehicles. Most of these vehicles rely on highly sophisticated and expensive sensors that are not yet feasible for commercial vehicles. On the other hand automated systems that are implemented in commercially available vehicles are largely limited to active safety scenarios where the system only assists the driver in dangerous situations such as collision avoidance or lane support.

The goal of this thesis project is to use sensors on commercially available vehicles for automating lateral control in selected scenarios. The evaluated scenarios are limited to roads where the sensors can detect lane markings or a preceding vehicle or both. The approach was to generate two different reference paths, one from lane markings and one from the preceding vehicle information. The lane marking path is generated from filtering measurements from the lane detection cameras using a nonlinear Kalman filter. The preceding vehicle path is generated by fusing the radar and camera measurements of the preceding vehicle. In order to develop and tune the filters a detailed analysis was done on the sensor measurements collected for this project.

The implemented filters improve the current system in several ways. When the lane marking are lost for short period of time the prediction from the last measurement update can provide a reference while driving up to 40 meters. The path generated from the estimates of the preceding vehicle describe the trajectory which the vehicle has driven. This way a more accurate reference signal can be generated than using only the current position of the preceding vehicle, especially in turns and at long distances. By having two references paths the lateral control is more robust and an algorithm that takes the covariances of the estimation for both paths into account guarantees a smooth transition between them.

Keywords: Lateral control, autonomous vehicles, sensor fusion, nonlinear kalman filtering.

Acknowledgements

We would like to thank Volvo Group Truck Technologies for giving us the opportunity to carry out this master thesis, Lennart Svensson for being our examiner and giving us valuable input, Erik Nordin and Mansour Keshavarz for being our supervisors at Volvo Group. We would also like to thank our group within Vehicle Dynamics & Active Safety for the help and support throughout the project.

Arni Thorvaldsson, Vinzenz Bandi, Gothenburg, June 2015

Contents

List of Figures

List of Tables

1
Introduction

This chapter will give a brief overview of the thesis project and relevant topics. Problem statement, aims and objectives will also be presented.

1.1 Background and motivation

The availability of driver assistance systems and autonomous functions has increased rapidly for production vehicles in recent years. Current systems and functions are being developed for active safety and aim to assist the driver in selected driving situations. Examples of such systems are lane keeping assistance, collision warning, emergency braking and adaptive cruise control. To further develop the technology and step closer towards fully autonomous vehicles several research projects have been initiated. At Volvo Cars "Drive Me" is an autonomous driving project in which 100 self-driving cars will be used on selected roads in and around Gothenburg. The first car is expected to be on the roads by 2017. Another project SARTRE [1] seeks to improve comfort, safety and energy consumption by developing platooning solutions. A platoon can be explained as a group of vehicles that drive in a coordinated formation. In SARTRE the leading vehicle is manually driven while the following vehicles follow it fully autonomously. By using vehicle to vehicle (V2V) communication together with the vehicles sensors the gap between the vehicles can be kept small which leads to reduced fuel consumption and less traffic congestion. One of the main obstacles in recent projects and autonomous driving in general is how to deal with inaccurate and unreliable sensors.
Within the department of Advance Technologies and Research (ATR) at Volvo Group Trucks Technology (GTT) are several projects focused on self-driving trucks. In some of these projects the truck currently relies on a unprocessed sensor measurements for lateral control which can cause problems when the conditions for the sensor are not ideal. The simplest way to get a reference for a lateral control is to use the sensor signal directly. This method is far from optimal and there is a big room for improvement in how the sensor data are handled. A simple filter that smooth's the sensor measurements yields better results. An optimal way is to use filters based on Bayesian statistics. The most common technique to implement Bayesian filter is the Kalman filter, invented in 1950s by Rudolph Emil Kalman [2]. The Kalman filter uses a process model that predicts what the new measurement is expected to be. The estimate is then derived by using the prediction, the measurement and the uncertainty of both of them. When more than one sensor source is available, the optimal way is to derive the estimate from all of them. Also when the sensor

source is not available for a short period, the prediction can be used as reference. How to handle sensor data in a optimal way using all information available was the motivation for this master thesis.

1.2 Problem Statement

The performance of lateral control is directly connected to the characteristics and ability of the used sensor. The sensors that are commercially available today are either not accurate enough or not reliable under all circumstances to be used individually. In order to have a robust lateral control all available sensor sources need to be used optimally. In this project a radar sensor, camera sensor and vehicle state sensors will be fused together to generate path for robust lateral control.

1.3 Applications

The sensors used in this project are limited to devices that exist in commercially available vehicles. These sensors are not capable of providing enough information for lateral control in complex urban scenarios. Therefore only selected scenarios are taken into account. Most of the scenarios take place on highways or roads where few unexpected things can occur. The work done in this thesis will provide a good basis on how to handle sensor data for lateral control in different applications. Some of the applications that are currently under development at Volvo ATR are described here together with how the reference for lateral control can be obtained in each of them.

Autopilot
Autopilot is a autonomous driving system for highways. The driver is supposed to monitor the system and be prepared to take over the control if the system has technical problems or dangerous situations appear in front of the vehicle. The lateral control can be done by using observations of lane markings, position of preceding vehicle and/or Gps measurements together with a map.

Platooning
Platooning is when multiple vehicles drive in unison, both laterally and longitudinally. The leading vehicle is usually manually driven but can also be driven by an autopilot system. The following vehicles are driven autonomously with predefined longitudinal distances between each vehicle. Similar to autopilot the lateral control can be done using the same references but vehicle to vehicle communication can be added to ensure string stability within the platoon [3].

Queue Assist
In traffic jam the queue assist system can take over longitudinal and lateral control of the vehicle to relieve the driver. The lateral control can be done using observations of lane markings and position of preceding vehicle.

1.4 Aims and objectives

The aim of this thesis is to improve the robustness of vehicle lateral controller that currently is under development for several projects at Volvo GTT.
To fulfill the aim the following scientific question needs to be answered:
"Can multiple sensor input improve robustness of lateral control for automated driving on highways?"
The objectives are to collect measurement data from available sensors that can be used for lateral control, analyze the data and use the findings to develop a sensor fusion algorithm that makes use of the sensors optimally to generate reference for lateral control. The algorithm will be tested with simulations and on a test track.

1.5 Thesis outline

The report is split into 8 chapters. Following the introduction, chapter 2 contains background theory and related work.
In chapter 3 the production and prototype sensors considered in this thesis are listed. Each sensors functions are described and how they are used in this project.. Also each sensor main characteristics and outputs described. Chapter 4 describes the methods used for sensor analysis and the development of filters and algorithms. Chapter 5 contains a description of the algorithm implemented for testing on the test truck. Chapter 6 presents some relevant results from the data analysis and algorithm implementation. In chapter 7 the conclusion are presented and discussion of main findings. Finally, the future work is proposed in chapter 8, with improvements of the current algorithm and outlook on other methods to further improve lateral control.

2
Background

2.1 Sensor Fusion

Methods in how to handle sensor data by combining different sensors often together with process model is a term called sensor fusion. By adding more information about a quantity measured by sensors or predicted by process model a more accurate estimate on the true quantity can be achieved than relying on a single source of information

Sensor fusion is often referred to in probabilistic robotics [4]. A probabilistic algorithm uses the dynamics of the robot and the measurements of its sensors. The dynamics can be modeled with a process model and state transition distribution determined from the uncertainties in the state evolution. The sensors on the robot have a certain noise on the measurements which can be quantified and the measurement distribution found. Combining the transition distribution and the measurement distribution in a statistical way e.g. using recursive Bayesian filter the estimate of the actual state can be derived.

2.1.1 Bayesian filter

A Bayesian filter calculates posterior density of the state recursively which represents a complete statistical description of the state at each time instance [4]. That is an assumption that the Bayesian filter makes called Markov assumption. Markov assumption specifies that if the current state $\mathbf{x}_{k|k}$ is known then the past and the future data are independent, i.e. the current state is sufficient representation of the past states. The posterior density is calculated when a new measurement is available in two steps:

Time update

In the time update step the Bayesian filter calculates the predicted mean $\hat{\mathbf{x}}_{k|k-1}$ and the predicted covariance $\mathbf{P}_{k|k-1}$ using the process model and the control input. The process model is often called the prediction model. The prediction model is usually a set of mathematical equations of the state evolution or how the systems states change from one time instance to the next one. The uncertainties of the state evolution come from the fact that modeling the dynamics accurately is often difficult and not computationally efficient and therefore approximations have to be made.

Measurement update

The update model calculates the mean $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ by using the prediction of the state, the measurements and their distributions. The sensors and the states are connected through the measurement model which describes how the sensors observe the states.

2.1.2 Kalman Filter

Kalman filter is the most common technique to implement a Bayesian filter. It was developed by Rudolph Emil Kalman [2] and published in 1960. The Kalman filter uses moments representation of Gaussian distribution. The estimates at sample k is represented with the mean $\hat{\mathbf{x}}_k$ and the covariance $\mathbf{P}_k$, in probabilistic terms the model is $p(x_k|y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$. To make sure that the correct posterior is calculated and it is Gaussian following properties have to hold:

- The initial belief at x_0 must be normally distributed.
- Kalman filter assumes linear dynamics thus the state transition probability $p(x_k|u_k, x_{k-1})$ must be linear function with additive Gaussian noise:

$$x_k = A_k x_{k-1} + B_k u_t + q_k \tag{2.1}$$

  where x_k is the state vector, A_k is the state transition matrix and q_k is the additive Gaussian noise that represents the uncertainty in the state transition.

- The measurement probability $p(y_k|x_k)$ must be linear with Gaussian added noise:

$$y_k = C_k x_k + r_k \tag{2.2}$$

  where y_k is the measurement vector, C_k is the measurement matrix and r_k is the added Gaussian noise.

Systems that fulfill these properties are called linear Gaussian systems.
The Kalman filter algorithm can be seen in equation 2.3 and 2.4. In equation 2.3 the state transition is implemented where the estimate prediction $\hat{\mathbf{x}}_{k|k-1}$ and the covariance $\mathbf{P}_{k|k-1}$ is calculated before incorporating the measurements.

$$\begin{aligned} &\textbf{Prediction}: \\ &\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \\ &\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \end{aligned} \tag{2.3}$$

The update step is shown in equation 2.4 where $\mathbf{K}_k$ is the Kalman gain, $\mathbf{v}_k$ is the innovation, $\mathbf{S}_k$ is the innovation covariance, $\hat{\mathbf{x}}_{k|k}$ is the new estimate and $\mathbf{P}_{k|k}$ is the covariance. The innovation checks how much new information is in the measurement and the Kalman gain determines how much the new measurement should be incorporated into the new state estimate. The moments $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ are then updated.

$$
\begin{aligned}
&\textbf{Update}: \\
&\mathbf{K}_k = \mathbf{P}_{k|k-1} - \mathbf{H}^T\mathbf{S}_k^{-1} \\
&\mathbf{v}_k = y_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1} \\
&\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k \\
&\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k \\
&\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T
\end{aligned}
\tag{2.4}
$$

When considering linear Gaussian dynamic systems, the Kalman filter provides optimal solutions for estimating the states in minimum mean squared error sense and is computationally efficient. But often systems are nonlinear and not Gaussian and therefore it is not possible to obtain closed form solutions for the estimates. In those cases linear Kalman filter is not applicable and a nonlinear filter is needed that makes approximation in order to derive the estimates.

2.1.3 Cubature Kalman Filter

Cubature Kalman filter(CKF) uses a spherical cubature rule for the approximation of an Gaussian filter [5] and can therefore be applied on nonlinear systems. The CKF is derivative free, which means there is no need to find the jacobian or hessian and would then not have problems with divergence.
In the CKF prediction step (see equation (2.5)) 2n sigma points $\mathcal{X}$ are generated where n is the number of states. The sigma points are then propagated through the process model in order to calculate the moments. The moments are the predicted mean $\hat{\mathbf{x}}_{k|k-1}$ and the predicted covariance $\mathbf{P}_{k|k-1}$.

$$
\begin{aligned}
&\textbf{Prediction}: \\
&\sigma - points: \\
&\mathcal{X}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{n}(\mathbf{P}_{k-1|k-1}^{(1/2)})_i \quad i = 1, 2, ...n \\
&\mathcal{X}_{k-1}^{(i+n)} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{n}(\mathbf{P}_{k-1|k-1}^{(1/2)})_i \quad i = 1, 2, ...n \\
&predicted\ moments: \\
&\hat{\mathbf{x}}_{k|k-1} \approx \frac{1}{2n}\sum_{i=1}^{2n} f(\mathcal{X}_{k-1}^{(i)}) \\
&\mathbf{P}_{k|k-1} \approx \frac{1}{2n}\sum_{i=1}^{2n} f(\mathcal{X}_{k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T + \mathbf{Q}_{k-1}
\end{aligned}
\tag{2.5}
$$

In the update step (see equation (2.6)) new sigma points are generated using the predicted moments. The sigma points are then propagated through the measurement model and the predicted measurements $\hat{\mathbf{y}}_{k|k-1}$ are found. The innovation covariance matrix $\mathbf{S}_k$ and the cross covariance matrix $\mathbf{P}_{xy}$ are estimated. Finally the estimate of the state are updated in the mean $\hat{\mathbf{x}}_{k|k}$ and the covariance $\mathbf{P}_{k|k}$. Note that the $\mathbf{P}_{xy}\mathbf{S}_k^{-1}$ is the Kalman gain.

Update :

$$
\begin{aligned}
&\sigma - points: \\
&\mathcal{X}_k^{(i)} = \hat{\mathbf{x}}_{k|k-1} + \sqrt{n}(\mathbf{P}_{k|k-1}^{(1/2)})_i \quad i = 1, 2, ...n \\
&\mathcal{X}_k^{(i+n)} = \hat{\mathbf{x}}_{k|k-1} - \sqrt{n}(\mathbf{P}_{k|k-1}^{(1/2)})_i \quad i = 1, 2, ...n \\
&desired\ moments: \\
&\hat{y}_{k|k-1} \approx \frac{1}{2n}\sum_{i=1}^{2n} h(\mathcal{X}_k^{(i)}) \\
&\mathbf{P}_{xy} \approx \frac{1}{2n}\sum_{i=1}^{2n} (\mathcal{X}_k^{(i)} - \hat{\mathbf{x}}_{k|k-1}) h(\mathcal{X}_k^{(i)} - \hat{y}_{k|k-1})^T \\
&\mathbf{S}_k \approx \frac{1}{2n}\sum_{i=1}^{2n} (h(\mathcal{X}_k^{(i)} - \hat{y}_{k|k-1}))(\cdot)^T + \mathbf{R}_k \\
&\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{xy}\mathbf{S}_k^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \\
&\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{xy}\mathbf{S}_k^{-1}\mathbf{P}_{xy}^T
\end{aligned}
\tag{2.6}
$$

2.1.4 Rauch-Tung-Striebel Smoother

The Rauch-Tung-Striebel (RTS) smoother is a smoothing algorithm for linear and Gaussian systems and was develop by Rauch, Tung and Striebel in the 1960s [6]. The RTS smoother is a forward-backward smoother which means it filters forward using Kalman filter (see section 2.1.2) computing $p(x_k|y_{1:k})$ for $k = 1, 2, 3...K$ and smooth's backward computing $p(x_k|y_{1:K})$ for $k = K-1, K-2, K-3...1$. The predicted moments and moments have to be saved for each time instance in the Kalman filtering step. The probabilistic model for RTS smoothing is $p(x_k|y_{1:K}) = \mathcal{N}(x_k; \hat{x}_{k|K}, P_{k|K})$ and the algorithm can be seen in equation 2.7

$$
\begin{aligned}
&\mathbf{G}_k = \mathbf{P}_{k|k}\mathbf{A}_k^T\mathbf{P}_{k+1|k}^{-1} \\
&\hat{\mathbf{x}}_{k|K} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k(\hat{\mathbf{x}}_{k+1|K} - \hat{\mathbf{x}}_{k+1|k}) \\
&\mathbf{P}_{k|K} = \mathbf{P}_{k|K} - \mathbf{G}_k(\hat{\mathbf{P}}_{k+1|K} - \hat{\mathbf{P}}_{k+1|k})\mathbf{G}_k^T
\end{aligned}
\tag{2.7}
$$

2.2 Control Theory

2.2.1 Feedback

When controlling the lateral position of a vehicle, the most frequently used approach is to control the lateral error Δy at a look ahead distance L to zero [7] [8]. One way of obtaining Δy is to estimate the lateral deviation between the road and the center of gravity (CG) of the vehicle (Δy_r) and the yaw angle with respect to the road (ε_r)and extrapolating to L ($\Delta y = \Delta y_r + L \cdot \varepsilon_r$) [9]. In [10] ε is used instead of ε_r to calculate Δy when DGPS provides the lateral reference. The equation to find ε becomes $\varepsilon = \varepsilon_r + \beta$ where β is the slip angle of the vehicle, which means that ε

is the angle of the velocity vector with respect to the road. These methods do not take the property of the path into account. If the path is available (e.g. when using a map or path planning) Δy can be defined as the lateral deviation from the path at the look ahead distance. In addition to the yaw angle, the yaw dynamics $\dot{\varepsilon}_r$ and $\dot{\beta}$ can be used to predict the lateral displacement caused by the curve the vehicle is taking [11]. Vision based systems can measure Δy at a look ahead distance directly. The lateral deviation Δy is used in the feedback control of the lateral controller. The look ahead distance L also has influence on the controller, a large L maintains stability of the control and dampens yaw dynamics, whereas at slow speeds a large L can cause cutting of sharp corners.

2.2.2 Feedforward

In addition to a feedback control, the controller performance can be improved by adding a feedforward component. Tai and Tomizuka noted the analogy between a vehicle lateral control system on curved road an a mechanical system with coulomb frictions, and proposed feedforward compensation based on road curvatures to improve the tracking error without compromising too much of other system performances [12] [9]. The feedforward control is usually based on the road curvature and the vehicle model. To estimate the road curvature [8] implemented an observer where the road curvature is included in the state vector. In Hasegawa and Konaka paper [13], they propose a method to estimate road curvature with three look-ahead points. As this is done in simulation it does not offer a method on how to obtain these look-ahead points in a real world application.

3

Sensors

The trucks used by the Advanced Technology and Research (Atr) department at Volvo Group are equipped with a variety of sensors that can be used in this project. This chapter describes the different sensors, their output signals and how they are used to generate a lateral control signal. A variety of different sensors were analyzed. These sensors include radar and camera systems for object detection as well as lane detectors, Gps and vehicle state sensors. Figure 3.1 shows a visualization of different measurements from sensors that are mounted on a truck.

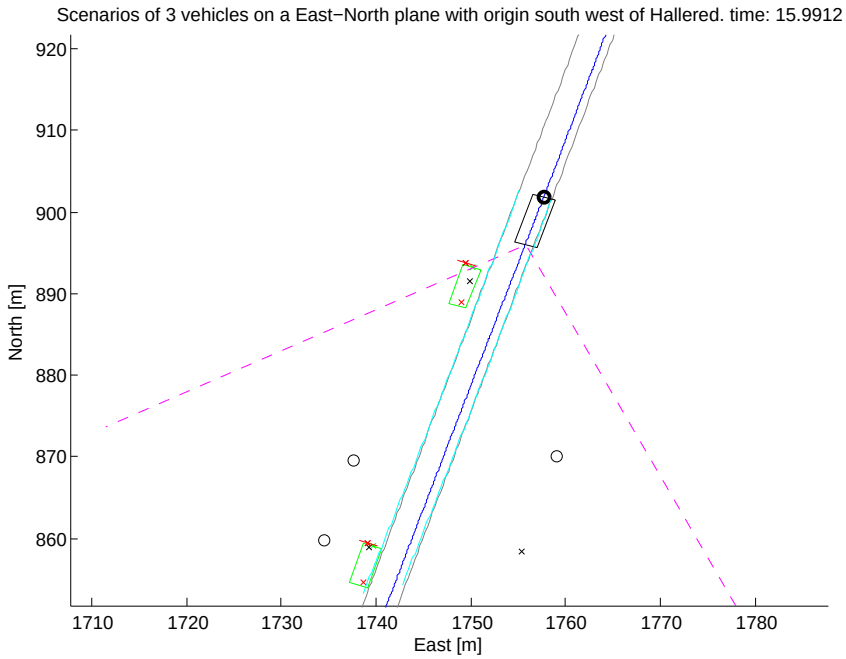


Figure 3.1: Visualization of ego vehicle and the two target vehicles with these signals. (black x,o) radar measurements. (green) position of the target vehicles from the Dgps System. (blue) path taken by the ego vehicle. (gray) lane markings. (magenta) field of view of the radar and the camera. (red) estimated positions (x on red line) of the vehicles from fused radar and camera measurements including position of the rear corners (connected by red line) and the far corner(x). (cyan) lane marking estimation from the camera.

3.1 Object Detection

The trucks have different sensors that are able to detect objects in the surrounding of the truck. The evaluated sensors include radar and camera. All of the evaluated sensors are mounted in front of the truck looking forward.

3.1.1 Radar

Radars can detect different objects in front of the EGO vehicle. The radar systems used in the automotive industry outputs preprocessed data from the radar measurements. The systems evaluated in this project can detect and track objects in their field of view and classify those. Measurements include the position and velocity of tracked objects in the EGO vehicle coordinate system (VCS). Classifications can range from "moving" or "stationary" to more sophisticated classifications such as different kinds of vehicles and stationary objects depending on their size and radar reflectivity.

3.1.2 Camera

Cameras have become increasingly important sensors in the automotive industry. With more sophisticated image processing algorithms and more powerful processors, cameras are very suitable for object detection. Similar to the radar the cameras used in the automotive industry detect different vehicles and objects in the vicinity of the EGO vehicle. The image processing algorithms detects and follows different objects and classifies them. More modern cameras also detect traffic signs or traffic lights as well as turn and brake signals of preceding vehicles.

3.2 Lane Detection Systems

Lane marking measurement systems are based on cameras and detects lane markings in the field of view of the camera. The measurements from such systems describe lane markings in the EGO VCS. The lane marking measurements can either be represented as clothoids ($\mathbf{z} = [\Delta y, \alpha, c_0, c_1]^T$) where every lane marking measurement has a lateral deviation from the vehicle (Δy), a heading (α), a curvature (c_0) and a curvature rate (c_1). Or the measurements are represented as polynomials in the VCS with the following form.

$$y(x) = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3 \tag{3.1}$$

Clothoid lane markings can be approximated to third order polynomials by using the following approximation.

$$y(x) = \Delta y - \alpha \cdot x + \frac{c_0}{2}x^2 + \frac{c_1}{6}x^3 \tag{3.2}$$

3.3 GPS

3.3.1 RACELOGIC VBOX System

For test purposes VBOX data loggers by RACELOGIC are available and can be mounted on the vehicles. The RACELOGIC VBOX is a GPS data logger that can be used in different setups. To use the system the vehicles have to be equipped with two GPS antennas and radio communication antennas. When one VBOX system is used as stand alone system absolute GPS coordinates can be logged with an accuracy of 2m. When two vehicles are equipped with a VBOX system each (moving base setup), the relative positions between the vehicles can be detected with an accuracy of ±2cm while the absolute positions remain at a 2m accuracy. The VBOX can also be used in conjunction with a stationary base station. With the base station the absolute position of a vehicle can be logged with an accuracy of ±2cm. The use of the bases station is limited by the radio communication range between the vehicle and the station which is a few hundred meters.

3.4 Sensor Setup

The goal for this project was to develop an application that could be used on a commercially available vehicle. Therefore the sensors used as inputs to the developed algorithms are all systems that are available in commercial vehicles. The following sensor systems have been used in the development of the algorithms.

- radar for object detection
- camera for object detection
- camera for lane detection
- vehicle state sensors for dead-reckoning

Apart from the sensors used in the algorithms a GPS system was used was used for generating ground truth data to test the algorithms.

3.4.1 Object Detection

Two different systems for object detection were used in this project, a radar based system and a camera based system. The radar system can detect and track six different objects in its field of view and classify them into stationary or moving. The measurements include object position and velocity in the EGO VCS including the standard deviations for the measurements. In addition to the radar a camera system is used that can detect six different objects and measure the objects positions in the EGO VCS. The data from the camera and the radar are fused together to find an accurate estimated position of the preceding vehicle. This is described in more detail in section 4.5. A statistical analysis of the sensor data can be found in section 6.4.

3.4.2 Lane Detection

The lane detection system used in this project is capable of measuring two different lane markings, one on each side of the vehicle. The lane markings are measured in the clothoid representation. The measurements and approximations of the left and right lane markings can be seen in equation (3.3) where x_v is the position on the x-axis of the Vcs.

$$
\begin{aligned}
&Measurements: \\
&\mathbf{z}^l = [\Delta y^l, \alpha^l, c_0^l, c_1^l]^T \\
&\mathbf{z}^r = [\Delta y^r, \alpha^r, c_0^r, c_1^r]^T \\
&Approximations: \\
&y^l(x_v) = \Delta y^l - \alpha^l \cdot x_v + \frac{c_0^l}{2} x_v^2 + \frac{c_1^l}{6} x_v^3 \\
&y^r(x_v) = -\Delta y^r - \alpha^r \cdot x_v + \frac{c_0^r}{2} x_v^2 + \frac{c_1^r}{6} x_v^3
\end{aligned}
\tag{3.3}
$$

The functions $y^l(x)$ and $y^r(x)$ describe the lateral position of the lane marking as a function of x_v in the Ego Vcs. The sign of Δy^r is negative due to the property of the camera. The measurement is positive although the lateral deviation of the right lane marking is negative in the Vcs. The heading angle α is the heading between the lane marking and the vehicle being positive in counter clock wise direction (i.e. it has the opposite to the heading in the Ego Vcs), therefore the coefficient is negative sign in equation (3.3). The coefficients c_0 and c_1 are are positive for curvatures and curvature rates in counter clock wise direction. For each measurement of a lane marking the camera generates a confidence value between 0 and 10. The measurements are used to generate a path in the middle of the two detected lane marking. The path is used to generate a lateral reference signal that the vehicle controller is supposed to follow. The filter algorithm with a Ckf prediction and a linear Kalman update is described in more detail in section 4.3. A statistical analysis of the sensor data can be found in section 6.1.

3.4.3 Vehicle State Sensors

All the trucks are equipped with a variety of of internal sensors that monitor the states of the vehicle. The measurements of these sensors are available on the internal Can (Controller Area Netork) buses of the trucks. In the algorithms developed within this project, the vehicle state sensors are used for dead-reckoning. Two different sensors are used for this part of the algorithms. The velocity measurement from the tachometer is used to estimate the distance the vehicle has traveled. The yaw rate measurement from the gyroscope is used to estimate the the vehicles rotation. In the filters described in chapter 4 the dead reckoning algorithm is explained in more detail.

3.4.4 GPS

The Vbox system is used in generating ground truth data for the lane marking measurements. To generate those the Gps data are combined with the lane marking

measurements, filtered with a Kalman filter and smoothed with a RTS-smoother. This is described in section 4.1. In all the measurements done for this project the VBOX system was configured in a moving base setup without stationary base station.

4
Methods

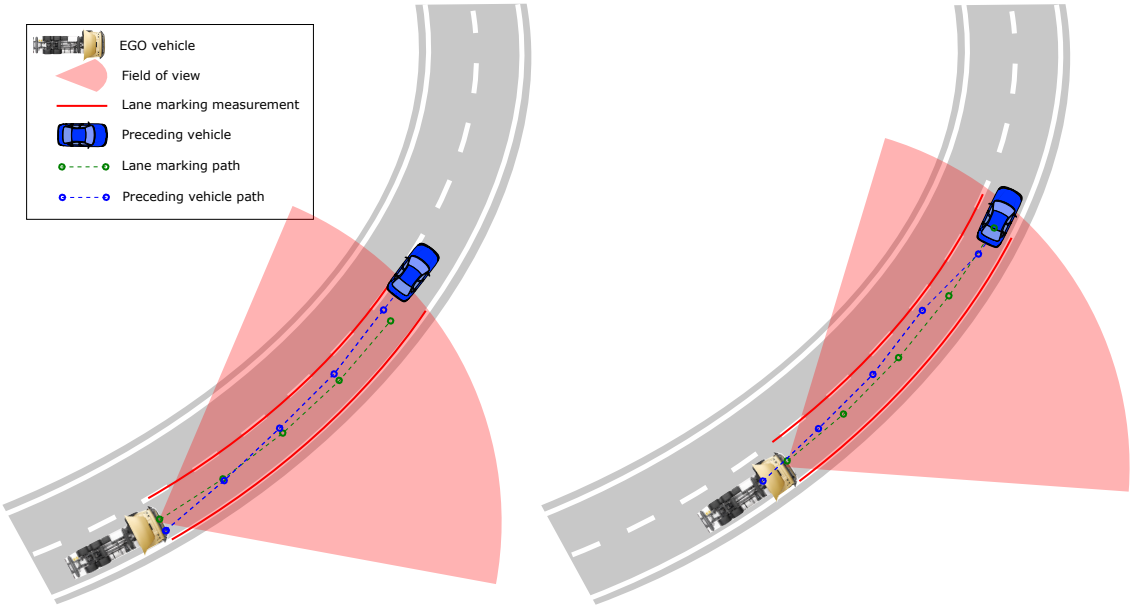


Figure 4.1: Overview figure that illustrates the sensor measurements and the paths that are generated using methods implemented in this thesis.

4.1 Generating a reference with lane markings

Lane markings can be used to generate a path on the road that the vehicle can follow by filtering the lane marking measurements. There are two main methods in which lane markings are represented by lane detection cameras. In the first method the camera outputs a third order polynomial that describes the lane marking in the vehicle coordinate system. The vehicle coordinate system has its x-axis in the direction of the heading of the vehicle.

$$y(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 \tag{4.1}$$

The second method describes the lane marking as a clothoid with a lateral deviation (Δy), a heading (α), a curvature (c_0) and a curvature rate (c_1). A lane marking described as a clothoid can be approximated with the following third order polynomial [14].

$$y(x) = \Delta y + \alpha \cdot x + \frac{c_0}{2}x^2 + \frac{c_1}{6}x^3 \tag{4.2}$$

The lane detection camera used in this project describes the lane markings as clothoids. It is able to detect two lane markings, one on the left side and one on the right side of the vehicle. Because the way the camera provides the coefficients, the exact polynomials for the left and right lane marking measurements look as follows.

$$\begin{aligned} y^l(x) &= \Delta y^l - \alpha^l \cdot x + \frac{c_0^l}{2}x^2 + \frac{c_1^l}{6}x^3 \\ y^r(x) &= -\Delta y^r - \alpha^r \cdot x + \frac{c_0^r}{2}x^2 + \frac{c_1^r}{6}x^3 \end{aligned} \tag{4.3}$$

4.1.1 Generating a path from lane markings

With the measurement of two lane markings a path in the center of the lane can be generated for the vehicle to follow. For practical purposes the path can also be offset from the center of the lane if this is desired, adding a constant to the coefficient Δy^p shifts the path parallel.
The lane marking measurements are in the form of $\mathbf{z}^l = [\Delta y^l, \alpha^l, c_0^l, c_1^l]^T$ and $\mathbf{z}^r = [\Delta y^r, \alpha^r, c_0^r, c_1^r]^T$ where $\mathbf{z}^l$ and $\mathbf{z}^r$ are measurements from the left lane and right lane respectively. In addition to the measurement the used lane detection camera provides a confidence value (conf^l, conf^r) between 0 and 10 for each measurement. Before a filter is applied the measurements from left and right land are combined to the path $\mathbf{z}^p = [\Delta y^p, \alpha^p, c_0^p, c_1^p]^T$ which describes the path in the middle of the lane. There are the following cases to be considered when calculating $\mathbf{z}^p$

Both lanes are detected
If both lane markings are detected the average between the coefficients can be calculated. If conf^l is equal to conf^r then $\Delta y^p = \Delta y^l - \Delta y^r$. The sign of Δy^r is negative due to the property of the camera. The measurement is positive although the lateral deviation of the right lane marking is negative in the Vcs. The other coefficients are then averaged. If conf^l and conf^r are not equal then Δy^p is calculated the same way but the other coefficients are the weighted averages of the respective coefficients in $\mathbf{z}^l$ and $\mathbf{z}^r$ where conf^l and conf^r are used as the weights.

One lane marking is not detected for a short time
In this case the path in the center of the lane markings can only be generated if the width of the lane is known. The width of the lane can be measured during the time when both lane markings are detected and it can be assumed that the lane width does not change over a short period of time. The coefficients $[\alpha^p, c_0^p$ and $c_1^p]^T$ are equal to the coefficients from the detected lane marking.

Only one lane marking is detected
In this case the width of the lane has to be assumed and the path can follow the lane marking at a predefined distance. The coefficients α^p, c_0^p and c_1^p are the same as from the detected lane marking.

No lane marking is detected
The measurement is discarded.

4.1.2 Coefficient and Sampled Lane Marking Measurement and Filters

There are two different ways to represent and filter a path generated from lane markings. One way is to have a coefficient representation like the ones in equations (4.1) and (4.2) where the lane marking function $y(x)$ has a value for all values x between $-\infty$ and $+\infty$. This representation describes the lane markings between certain bounds on x between 0 and around 50 m. The second way is to represent the path as sampled points in x and y similar to the methods proposed in Fernándes et.al. paper [15]. The following chapters describe both principles in more detail. For both methods a filter method is presented. Ultimately it was decided to go forward with the sampled version as it is less sensitive to coefficient noise [14] and the performance of the sampled filter has proven to be better in tests than the coefficient filter (see 6.2).

4.2 Coefficient Filter for Lane Marking Path

The first filter is a coefficient filter which is a Kalman filter with both linear process and measurement model. The development of this filter was discontinued after initial test showed that the sampled filter described in section 4.3 performed better as seen in section 6.2.

4.2.1 Prediction step

Using the third order polynomial that approximates a clothoid in equation 4.2 the motion model can be derived. The states that are predicted with the motion model are the lateral deviation (Δy), the heading (α), the curvature (c_0) and the curvature rate (c_1) of the averaged lane marking. The movement and change in orientation of the truck is calculated using measurements of the velocity(v), yaw rate ($\dot{\varphi}$) and time between lane marking measurements(dt). The distance traveled between each prediction is then $x = v \cdot dt$ and the change in orientation is $\frac{dx}{dy} = atan(-\dot{\varphi} \cdot dt)$. Since the angles are very small during highway driving, small angle approximation can be applied which gives $d\varphi = -\dot{\varphi} \cdot dt$. The prediction model can be seen in

equation 4.4, x_v is the position on the x-axis of the VCS.

$$\begin{aligned}
&\mathbf{x}(k+1) = A \cdot \mathbf{x}(k) + B \cdot u \\
&where: \\
&\mathbf{x}(k) = \begin{bmatrix} \Delta y \\ \alpha \\ C_0 \\ C_1 \end{bmatrix} \\
&u = \dot{\varphi} \\
&A = \begin{bmatrix} 1 & x_v & \frac{x_v^2}{2} & \frac{x_v^3}{6} \\ 0 & 1 & x_v & \frac{x_v^2}{2} \\ 0 & 0 & 1 & x_v \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&B = \begin{bmatrix} 0 \\ -dt \\ 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{4.4}$$

4.2.2 Update step

Since all the states are measured the update is linear with the measurement matrix H as identity matrix. When the confidence value (conf^l, conf^r) are both below confidence threshold (set to 3) the measurement are considered not valid and in that case the update step is skipped.

$$\begin{aligned}
&\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(y_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \\
&\mathbf{H} = \mathbf{I}_n \\
&\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \\
&\mathbf{K}_k = \mathbf{P}_{k|k-1} - \mathbf{H}^T\mathbf{S}_k^{-1} \\
&\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k \\
&\mathbf{R}_k : \textit{Measurement noise covariance}
\end{aligned} \tag{4.5}$$

4.3 Sampled Filter for Lane Marking Path

The sampled filter is a Kalman filter with a nonlinear process model for the prediction and a linear measurement model for the update. For the prediction step a Cubature Kalman filter is used. Since the measurement model is linear the update is a direct Kalman update step. It was decided to go forward with this filter in this project as it performed better than the coefficient filter.

4.3.1 Prediction step

The prediction step of the filter estimates the positions of the sampled path at the time of the next lane marking measurements. It does that by applying the nonlinear

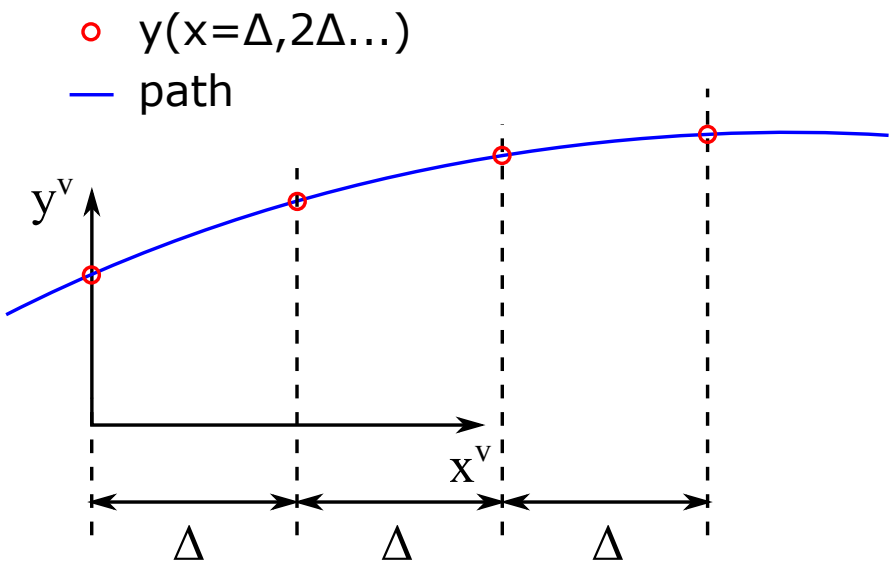


Figure 4.2: Shows how the path is sampled for the filter. Like the lane marking measurements the sampled path is described in the vehicle coordinate system (Vcs) (x^v, y^v). The samples are distributed equidistantly in x at distance Δ.

function $f(\mathbf{x}, \mathbf{v}, \dot{\varphi}, \mathbf{dt})$ to the previous estimation. All the formulas for the Cubature Kalman prediction can be found in (4.8).

Nonlinear prediction function $f(\mathbf{x}, \mathbf{v}, \dot{\varphi}, \mathbf{dt})$ The nonlinear function for the prediction step is divided in three steps.

1) Move coordinate system

Using the time elapsed since the previous measurement (dt), the yaw rate $(\dot{\varphi})$ and the velocity of the vehicle (v), the coordinate system (x^v_{k-1}, y^v_{k-1}) is moved to the new position (x^v_k, y^v_k). The coordinate system is first moved along the x-axis by $dx = v \cdot dt$ and then rotated around the z-axis by $d\varphi = \dot{\varphi} \cdot dt$ (Figure 4.3.

2) Update estimates

The estimates $\mathbf{x}_{k-1} = [y^0_{k-1}, y^1_{k-1}, ...]$ are moved to the new coordinate system. Since this transformation and a rotation the updated points p^i have both x and y values (Figure 4.4).

$$\begin{aligned} p^i_x &= (\Delta \cdot i - dx) \cdot cos(-d\varphi) - y^i_{k-1} \cdot sin(-d\varphi) \\ p^i_y &= (\Delta \cdot i - dx) \cdot sin(-d\varphi) + y^i_{k-1} \cdot cos(-d\varphi) \\ i &= 0, 1, ...n \\ n &: nr.\ of\ points\ in\ discretized\ path\ minus\ 1 \end{aligned} \tag{4.6}$$

3) Interpolation/Extrapolation

The prediction points are calculated by linearly interpolating from p^i to $x =$

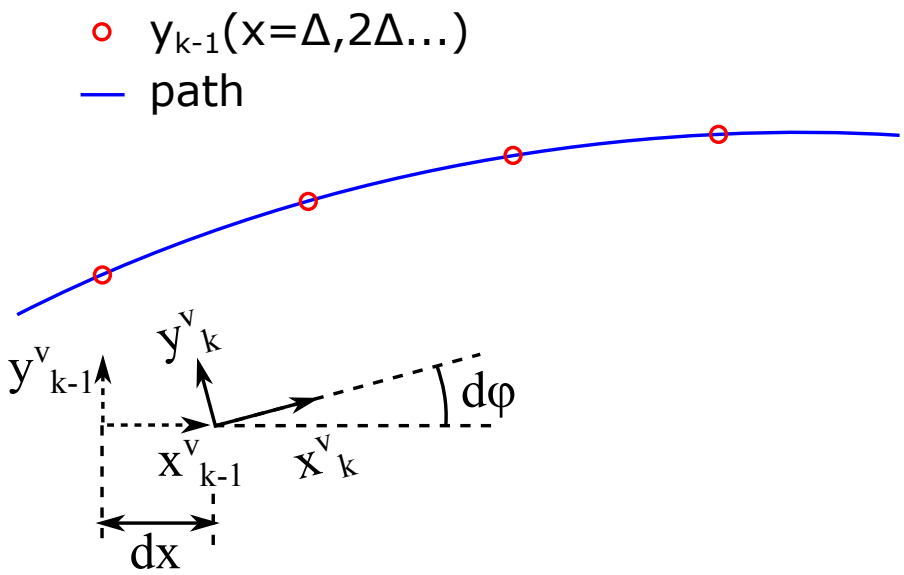


Figure 4.3: Moving the coordinate system in the nonlinear function $f(\mathbf{x}, \mathbf{v}, \dot{\varphi}, \mathbf{dt})$in the prediction step

$[0, \Delta, 2\Delta...]$. (Figure 4.5).

$$\begin{aligned} y_k^i &= p_y^i + ((\Delta \cdot i - p_x^i) \cdot (p_y^{i+1} - p_y^i))/(p_x^{i+1} - p_x^i) \\ i &= 0, 1..n-1 \\ y_k^n &= p_y^i + ((\Delta \cdot n - p_x^n) \cdot (p_y^n - p_y^{n-1}))/(p_x^n - p_x^{n-1}) \end{aligned} \tag{4.7}$$

Since it is not possible to interpolate a value for the last x value, the last point has to be extrapolated. In the current implementation this is done with a linear extrapolation through p^{n-1} and p^n. On alternative method would be to extrapolate by setting the angle $\sphericalangle p^{n-1}, p^n, p(\Delta \cdot n, y_k^n)$ equal to $\sphericalangle p^{n-2}, p^{n-1} p^n$

Since $f(\mathbf{x}, \mathbf{v}, \dot{\varphi}, \mathbf{dt})$ is nonlinear the prediction step is done with a Cubature Kalman filter.

$$\begin{aligned} &\mathcal{X}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{n}\left(\mathbf{P}_{k-1|k-1}^{(1/2)}\right)_i, i = 1, 2, ..., n \\ &\mathcal{X}_{k-1}^{(i+n)} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{n}\left(\mathbf{P}_{k-1|k-1}^{(1/2)}\right)_i, i = 1, 2, ..., n \\ &W_i = \frac{1}{2n}, i = 1, 2, ..., 2n \\ &n : nr.\ of\ points\ in\ discretized\ path \\ &\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{2n} f(\mathcal{X}_{k-1}^{(i)}, v, \dot{\varphi}, dt) W_i \\ &\mathbf{P}_{k|k-1} = \sum_{i=1}^{2n} f(\mathcal{X}_{k-1}^{(i)}, v, \dot{\varphi}, dt) - \hat{\mathbf{x}}_{k|k-1})(\cdot)^T W_i + \mathbf{Q}_{k-1} \end{aligned} \tag{4.8}$$

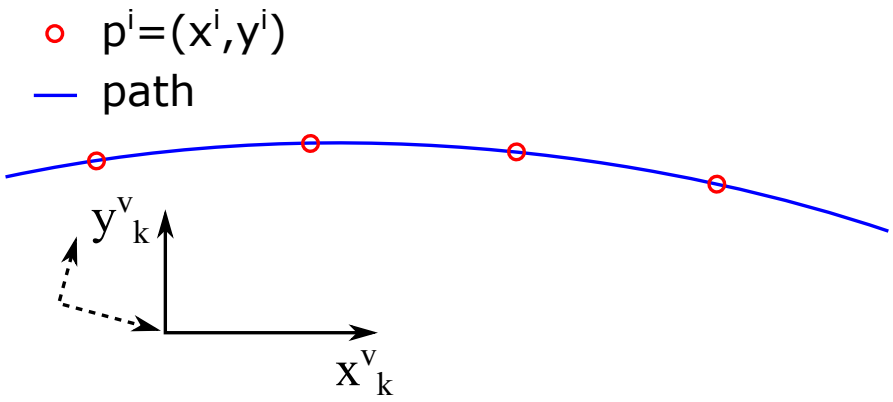


Figure 4.4: Update estimates to moved coordinate system in the nonlinear function $f(\mathbf{x}, \mathbf{v}, \dot{\varphi}, \mathbf{dt})$in the prediction step

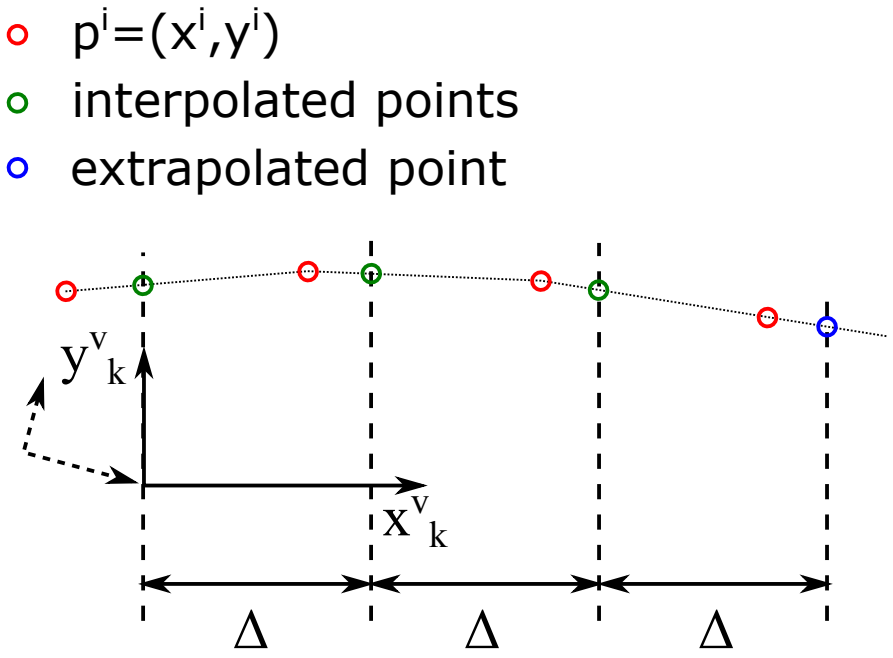


Figure 4.5: interpolation/extrapolation to obtain prediction points in the nonlinear function $f(\mathbf{x}, \mathbf{v}, \dot{\varphi}, \mathbf{dt})$in the prediction step

4.3.2 Update step

Measurement model This filter is set up in a way that the measurements are the sampled path. Therefore the lane marking estimations from the lane detection camera must be translated to sampled path points. The lane marking measurements are in the form of $\mathbf{z}^p = [\Delta y^p, \alpha^p, c_0^p, c_1^p]^T$ according to 4.1.1.

The path $\mathbf{z}^p$ can then be used to calculate the sampled measurement points for the filter.

$$y = \begin{bmatrix} 1 & \Delta & \frac{1}{2}\Delta^2 & \frac{1}{6}\Delta^3 \\ 1 & 2\Delta & \frac{1}{2}(2\Delta)^2 & \frac{1}{6}(2\Delta)^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & n\Delta & \frac{1}{2}(n\Delta)^2 & \frac{1}{6}(n\Delta)^3 \end{bmatrix} \mathbf{z}^p \tag{4.9}$$

The update step is then a linear Kalman update (see equation 4.10)

$$\begin{aligned} &\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(y_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \\ &\mathbf{H} = \mathbf{I}_n \\ &\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \\ &\mathbf{K}_k = \mathbf{P}_{k|k-1} - \mathbf{H}^T\mathbf{S}_k^{-1} \\ &\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k \\ &\mathbf{R}_k : \textit{Measurement noise covariance} \end{aligned} \tag{4.10}$$

4.4 Generating a reference with preceding vehicle

When a vehicle is driving in front of the Ego vehicle its position can be detected by the object detection sensors (radar and camera). These positions can be used as a reference for the lateral control of Ego vehicle.

4.4.1 Generating reference path

A reference path can be constructed from the filtered position of the preceding vehicle. The path contains N number of points in the Ego vehicles coordinate system. The points are placed equidistantly in the x-axis of the Vcs with a fixed longitudinal distance deltaPrec between them.

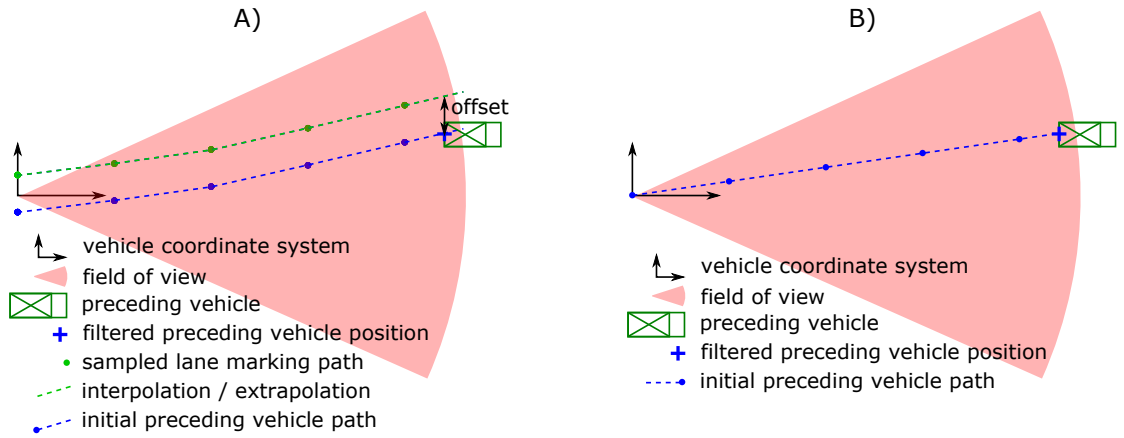


Figure 4.6: Initial path from preceding vehicle when a vehicle enters the field of view. A) shows how the N points in the path are generated if a lane marking path is available by offsetting the lane marking path and adding the preceding vehicle position as the Nth point and B) shows how the points are generated when no lane marking path is available by linear interpolation between the ego vehicle position and the preceding vehicle position.

Initial path generated

When there is no vehicle visible to the sensors no path is generated. As soon as a vehicle appears in the field of view of the Ego vehicle a path between the Ego vehicle and the preceding vehicle is generated depending if the lane markings path is available or not. The path is a series of N points in the vehicle coordinate system. If the lane markings are available the points are placed parallel to the lane markings with an offset. The offset is the lateral difference between the position of the preceding vehicle to the lane markings. The lane marking path consists of N-1 points in the vehicle coordinate system, therefore the Nth point in the preceding vehicle path is the position of the preceding vehicle. This method of generating the initial path can be seen in figure 4.6 A. If no lane marking path is available the initial points are placed by linear interpolating six points between the Ego vehicle and the preceding vehicle (figure 4.6 B).

Movement of the path

The points of the path represent the trajectory of the preceding vehicle in Ego vehicle coordinate system. Since the coordinate system moves with the Ego vehicle the points need to move within the coordinate system according to the distance traveled (dx) and change in orientation ($d\varphi$) of the Ego vehicle. The distance and the orientation is calculated using the yaw rate ($\dot{\varphi}$), the velocity measurement(v) and the step length(dt):

$$d\varphi = dt \cdot \dot{\varphi}$$

$$dx = dt \cdot v$$

Update points of the path

When the distance between the point in path which is closest to the preceding vehicle and the current filtered point on the preceding vehicle is equal to deltaPrec, oldest point (furthest away from the preceding vehicle) is discarded and current filtered point added to the path. The update is illustrated in figure 4.7.

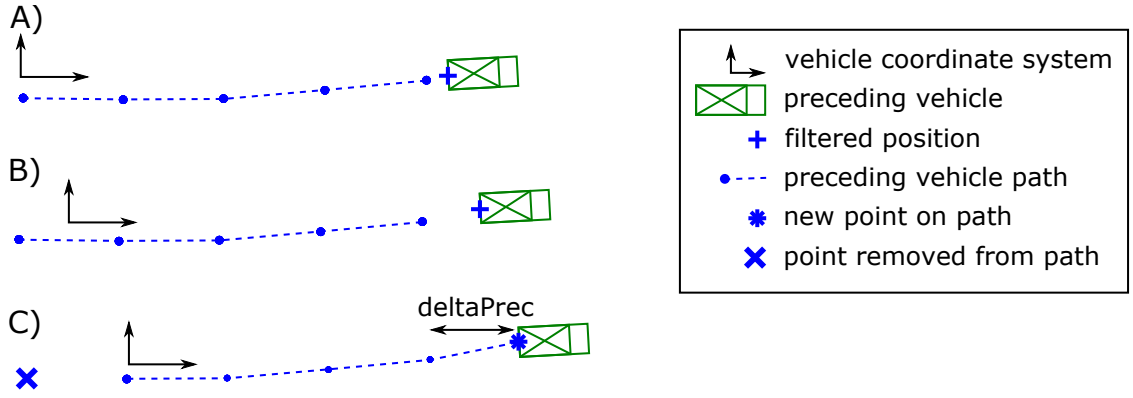


Figure 4.7: Updating the preceding vehicle path. Each time the distance between the newest point in the path and the preceding vehicle is larger than deltaPrec the point furthest away from the preceding vehicle is removed and the preceding vehicle position is added as new point.

Uncertainties

The movement of the points can be considered as pure prediction since there are no measurements of the points in the path to update with. The prediction step is the same as for the lane marking sampled points and therefore the same prediction model can be used. The covariance is based on the noise of the yaw rate and the velocity. Since there is no update step, the uncertainties increase over time, which results in very uncertain points on the path especially if the preceding vehicle is far ahead. By filtering the states of the vehicle such as vehicle and heading a more accurate movement of the vehicle between time instances can be achieved. This was not in the scope of this thesis but discussed in future work chapter 8.

4.5 Preceding Vehicle Filter

The position of the preceding vehicle is measured with the camera and radar. The camera measures longitudinal and lateral distances to the preceding vehicle relative to the Ego vehicle. The radar has the same measurements as the camera but also outputs the relative velocity in longitudinal and lateral direction. These sensor information can be fused together to obtain an estimate of the position of the preceding vehicle relative to Ego vehicle. Based on the measurements of the preceding vehicle the most straight forward choice of a state vector would be $[x, y, v_x, v_y]$ where x and y is the longitudinal and lateral distances respectively, v_x and v_y are the correspond-

ing velocities in longitudinal and lateral direction. All of the states are relative to the Ego vehicle.

4.5.1 Prediction step

When choosing a process model the changes between the measurements samples need to be considered. The measurements are relative to the Ego vehicle and the assumption can be made that the two vehicles travel on the same lane with similar velocity which makes the changes in velocity slow. Therefore a constant velocity (CV) model was chosen. Adding acceleration to the model would not be bring much improvements since the change in acceleration are very small. Since the CV process model is linear the optimal prediction method would be using the Kalman prediction. Since the measurements are not synchronous the step length dt might need to be adaptive, analysis on different step lengths can be seen in section 6.4. The Kalman prediction equations and a discretized CV model can be seen in equation4.11.

$$
\begin{aligned}
&\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} \\
&\hat{\mathbf{P}}_{k|k-1} = \mathbf{A}\hat{\mathbf{P}}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q}_k \\
&where: \\
&\mathbf{A} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\mathbf{Q_k} = \begin{bmatrix} \frac{dt^3}{3} & 0 & \frac{dt^2}{2} & 0 \\ 0 & \frac{dt^3}{3} & 0 & \frac{dt^2}{2} \\ \frac{dt^2}{2} & 0 & dt & 0 \\ 0 & \frac{dt^2}{2} & 0 & dt \end{bmatrix}
\end{aligned}
\tag{4.11}
$$

4.5.2 Update step

In the update step both the radar and the camera measure the x and y states. Different methods can be used to derive the estimate. The optimal method would be to update first using one sensor and then update with the second one. In equation 4.12 the radar and the camera update steps can be seen.

$$
\begin{aligned}
&\textbf{RadarUpdate} \\
&\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_{radar} \\
&\mathbf{K}_k = \mathbf{P}_{k|k-1} - \mathbf{H}^T\mathbf{S}_k^{-1} \\
&\hat{\mathbf{x}}_{k|k}^{Radar} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(y_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \\
&\mathbf{P}_{k|k}^{radar} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \\
&where: \\
&\mathbf{H} = \mathbf{I}_n \\
&\mathbf{R_{radar}} = diag(\begin{bmatrix}\sigma_{LonPosRad}^2 & \sigma_{LatPosRad}^2 & \sigma_{LonVelRad}^2 & \sigma_{LatVelRad}^2\end{bmatrix}) \\
&\textbf{CameraUpdate 2} \\
&\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k}^{radar}\mathbf{H}^T + \mathbf{R}_{camera} \\
&\mathbf{K}_k = \mathbf{P}_{k|k}^{radar} - \mathbf{H}^T\mathbf{S}_k^{-1} \\
&\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k}^{Radar} + \mathbf{K}_k(y_k - \mathbf{H}\hat{\mathbf{x}}_{k|k}^{Radar}) \\
&\mathbf{P}_{k|k} = \mathbf{P}_{k|k}^{radar} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \\
&where: \\
&\mathbf{H} = \begin{bmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0\end{bmatrix} \\
&\mathbf{R_{camera}} = diag(\begin{bmatrix}\sigma_{LonPosCam}^2 & \sigma_{LatPosCam}^2\end{bmatrix})
\end{aligned}
\tag{4.12}
$$

Since the radar and camera measurements are not synchronous there are few cases to consider dependent on which measurements are available when the filter is executed:

Radar and camera measurements available
The update step includes both updates according to equation 4.12.

Only radar measurement available
The update step runs update 1 from equation 4.12.

Only camera measurement available
The update step runs update 2 from equation 4.12.

No measurements available
Update step is skipped.

4.6 Control Signals

The reference paths from the lane marking and preceding vehicle are represented in same way. The control signals can therefore be calculated in same way for both paths. The control signals are lateral error and heading error of the truck relative to the lane at a look ahead distance (see figure 4.8). The look ahead distance is

dependent on the trucks velocity and a look ahead time ($lookAheadDistance = velocity \cdot lookAheadTime$), the look ahead time is set to 1 second. A decision has to be made which path to follow and that is done by using the uncertainty of the paths at the look ahead point.

Lane Marking Path Calculate the control signal from the lane marking path. If the uncertainty of the lane marking gets larger than uncertainty of preceding vehicle path change to the state "Preceding Path".

Preceding Path Calculate the control signal from the preceding vehicle path. If the uncertainty of the preceding vehicle path grows larger than uncertainty of lane marking path, change to the state "Lane Marking Path".

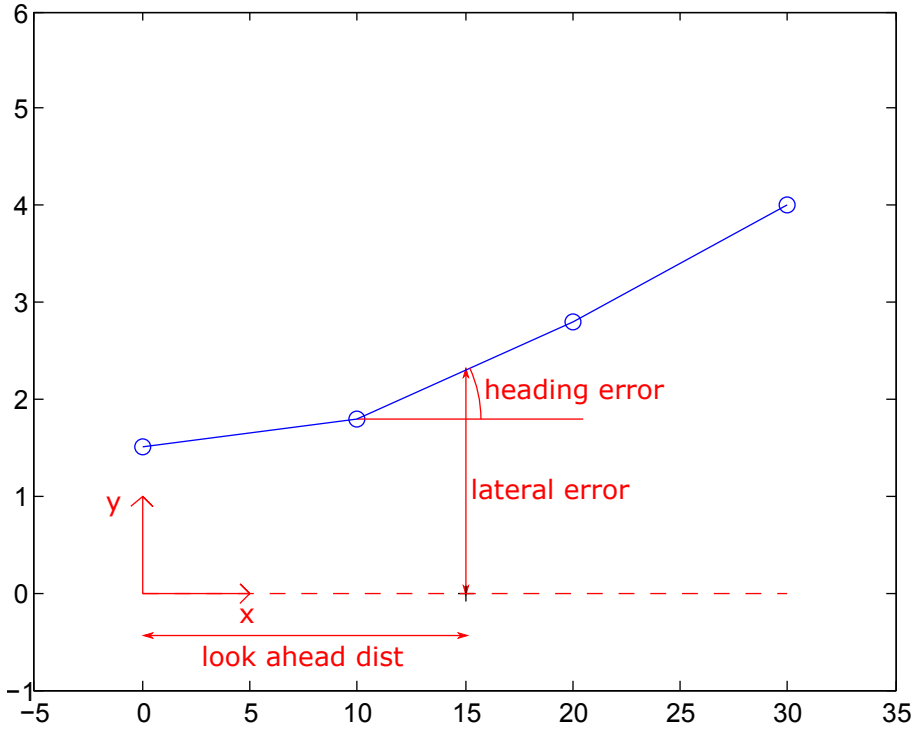


Figure 4.8: Lateral and heading error to the path obtained from the sampled filter estimates in the Vcs.

Another approach could be to fuse the two paths together obtaining a single path that can be used as reference. This approach was not chosen since the lane marking path is in the middle of the lane but the preceding path is varied from the middle. This is because it is difficult to keep a constant lateral position in a lane while driving. This can be seen in figure 6.14. Also the uncertainty of the lane marking is much smaller than the uncertainty of the preceding vehicle path when lane marking measurements are available.
In the case of coefficient filter the lateral error is calculated using the approximation in equation 4.2 and the heading error is then its derivative. The equations can be seen in equation 4.13.

$$
\begin{aligned}
y_{ref} &= \Delta y + \alpha \cdot x + \frac{c_0}{2}x^2 + \frac{c_1}{6}x^3 \\
\theta_{ref} &= \alpha + c_0 \cdot x + \frac{c_1}{2}x^2
\end{aligned}
\tag{4.13}
$$

4.7 Experimental Data

In order to tune the weights and coefficients in the filter a good sets of measurement data is vital. The data was logged on the AstaZero test track using Volvo FH truck as a subject vehicle and Volvo V60 as a target vehicle. A map of AstaZero can be found in appendix A.1.

4.7.1 Setup

The Ego vehicle contains the measurement computer and takes care of logging the data. The Vbox system in the target vehicle sends the Dgps data to the Vbox system in subject vehicle. The sensor information that is logged during the scenarios are listed in 4.1.

Sensor information	
Truck (Subject)	**Car** (Target)
Velocity	Dgps
Yaw rate	
Dgps	
Lane Marking	
Camera objects	
Radar objects	

Table 4.1: The sensor information logged during the scenarios

4.7.2 Scenarios

Description and purpose with different scenarios that were performed during logging are as follow:

Highway driving

Description: Driving on highway velocity (70 - 90 km/h) without any obstruction blocking the lane markings from the camera.
Purpose: The data will be used to tune the lane marking filters since the vision of the lane marking should be ideal.

Highway driving with preceding vehicle

Description: Driving on highway velocity with preceding vehicle visible to the object detection sensors. Both vehicle should be driving in same lane and the distance between them from 20 meters up to 50 meters.

Purpose: The sensor information from radar and camera on lateral and longitudinal distances of objects ahead will be used to develop filter that tracks the preceding vehicle.

Traffic jam

Description: The preceding vehicle will simulate a traffic jam by varying the velocity from standstill to normal highway driving.
Purpose: Further develop and tune the prediction step of the filter as well as handling the transition between different sensor sources. When the car is driving less than 7 m/s the camera does not detect the lane marking which needs to be taken care of by using pure prediction and the detection of the preceding vehicle. When the car is at standstill it needs to make sure that the filter covariance matrix does not increase.

One test run simulating highway driving is analyzed in more detail in section 4.8. Figure 4.9 (left) shows the trajectory of the truck during the measurement run.

4.8 Lane Marking Sensor Analysis

In the lane marking filter explained in section 4.1 the lane marking measurements are used for the measurement updates. To do the measurement update the measurement noise covariance matrix R is needed. Unfortunately this information is not provided by the manufacturer of the lane marking detection camera. To obtain the covariance matrix R measured data is compared to ground truth to obtain the distribution of the sensor measurement error. This analysis was done for the sampled measurements as described in section 4.1.2.

4.8.1 Ground Truth

To evaluate the measurements, the ground truth must be known. The data sets that are used to characterize the measurements were collected on the "rural road" on the AstaZero test track. The rural road in AstaZero is a closed loop with a length of around 5 km with turns in both directions. The tightest turn on the rural road is approximately 300 m in radius. The data sets were collected by driving around the rural road in counter clock wise direction. A map of AstaZero can be found in appendix A.1. Together with the lane markings the Gps position and heading of the Ego vehicle was collected during these measurements. With the Gps measurements of the lane markings can be mapped from the vehicle coordinate system (Vcs) to a map. In this project there were two possible maps, an existing AstaZero map or a map generated form the measured data.

AstaZero Map

At the Atr department at Volvo Gtt a detailed map with the measured positions of the lane markings at AstaZero is available. Unfortunately this map could not be used as ground truth in this project. The Gps measurement were collected without a base station for cm accuracy (see section 3.3). This means that the Gps measurements have an offset to the actual positions of

the Ego vehicle. It was not possible to map the measured Gps data onto the existing map with a high enough accuracy.

Generated from Measured Data

The second possibility to get ground truth information is to extract it from measurements by combing Gps measurements and lane marking measurements. Even though the absolute Gps positions are not exact, the relative Gps position in one measurement run is more accurate. This is because the bias on the measurements primarily depends on atmospheric effects and sky blockage. These are slow changing effects and can therefore be assumed to be constant during the short time of one measurement run. The distance information from the lane marking measurement (Δy) was used together with the Gps measurements. For a description of the lane marking measurements see section 3.2. The Gps measurements are transposed to a East-North-Up (Enu) coordinate system and each lane marking measurement is transposed to its corresponding vehicle position and heading. This way the two lane markings can be mapped on an Enu coordinate system. The obtained lane markings positions are subject to both the Gps measurement noise (not the bias from the atmospheric effects) and the measurement noise of the (Δy) measurement of the lane marking camera. The used data set was collected under very good conditions for the lane marking detection camera i.e. dry roads and no sunshine. The lane markings are filtered with a Kalman filter with a constant velocity model and smoothed with an Rts smoother. The path and cutout of the measured, filtered and smoothed lane marking can be seen in figure 4.9.

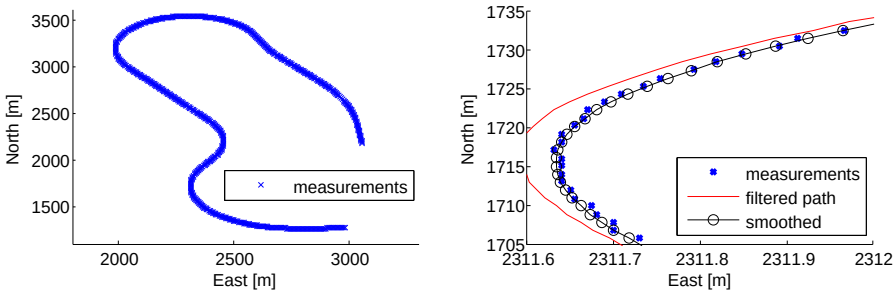


Figure 4.9: (left) plot of the trajectory the vehicle drove during the measurement run in an Enu coordinate system. (right) Cutout of the ground truth of the lane markings generated with Gps data, lane marking measurements using a filter and and Rts-smoother.

4.8.2 Evaluating the Measurements

For the evaluation each lane marking measurement was compared to the ground truth. The lane marking measurements are collected in the vehicle coordinate system (Vcs). For comparison the ground truth is transposed from Enu to the vehicle coordinate system using the vehicle position and heading (from Gps data) at each measurement instance. The measurements are sampled at $x = \{0m, 10m, 20m, 30m, 40m\}$.

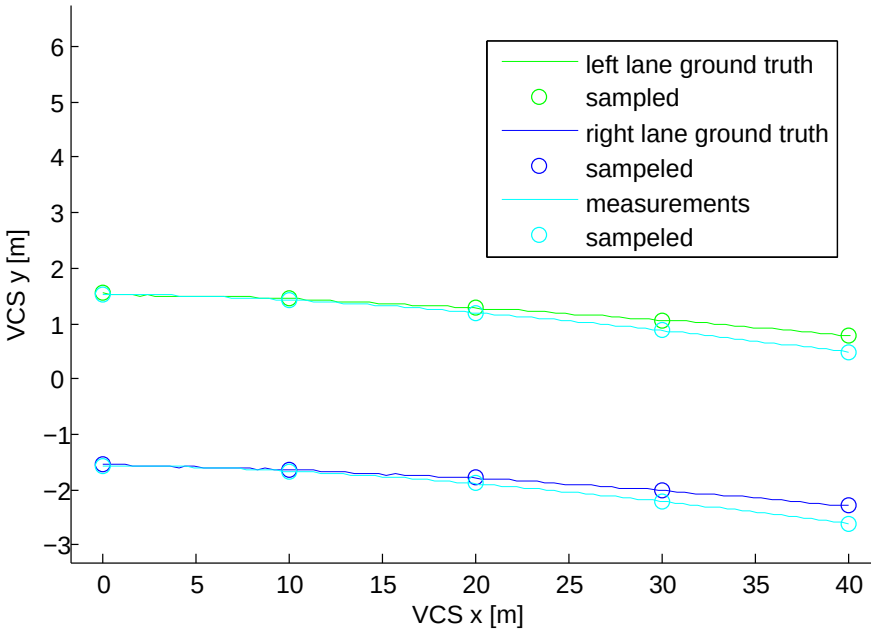


Figure 4.10: Visualization of the ground truth of the lane markings and a corresponding measurement in the VCS.

To calculate the error, the ground truth is interpolated to the same x positions as the measurement samples and subtracted from the measurement points. Figure 4.10 shows the ground truth lane markings and the measurements in the vehicle coordinate system. The error at the sampled x positions is analyzed for each lane marking measurement. The results for this can be found in section 6.3

4.9 Preceding Vehicle Sensor Analysis

Initial intention was to use the relative distances between the vehicles measured with the Vbox system as ground truth for filter implementation. The Vbox system was not working as it should and the data was not accurate enough to use as ground truth so another approach was taken.

4.9.1 Generating ground truth using RTS smoother

Since the actual ground truth of the preceding vehicle is unknown the best knowledge about it is from the measurements. To generate the ground truth the measurement are filtered using all the measurements samples available and have adaptive step length between each prediction. The moments are stored and the back smoothed using Rts smoother. The ground truth will then be used as reference when evaluating the step length used for implementation of the filter. The evaluation can be seen in section 6.4.

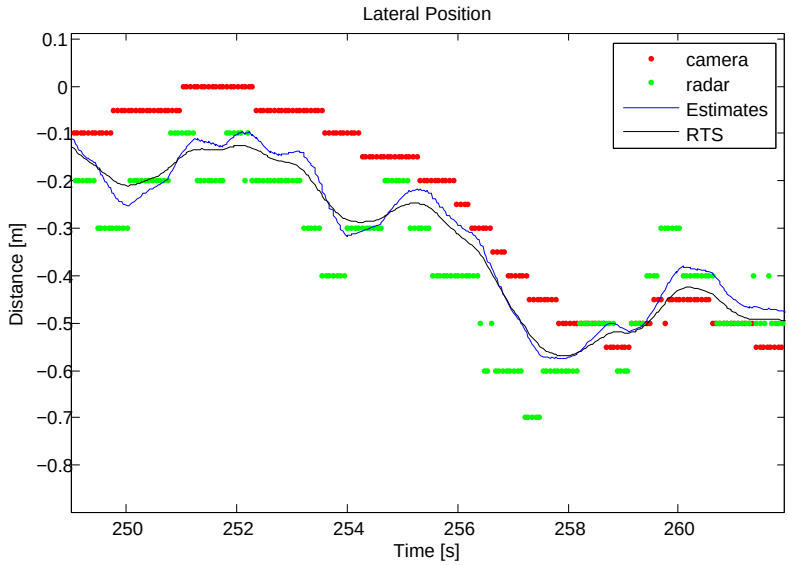


Figure 4.11: Lateral position of the preceding vehicle. The red and green dots are the camera and radar lateral measurements respectively. The blue line is the estimated lateral position and the black line is the smoothed position or the Ground Truth.

5

Implementation

5.1 Simulink Implementation

The algorithm proposed in this thesis is implemented in Simulink for running on the test truck at Volvo Group Advanced Technology & Research Atr department. Since the algorithm is developed in Matlab the Simulink implementation contains the main processing blocks of the algorithm as Matlab function blocks. Figure 5.1 shows the block diagram of the algorithm implemented in Simulink.

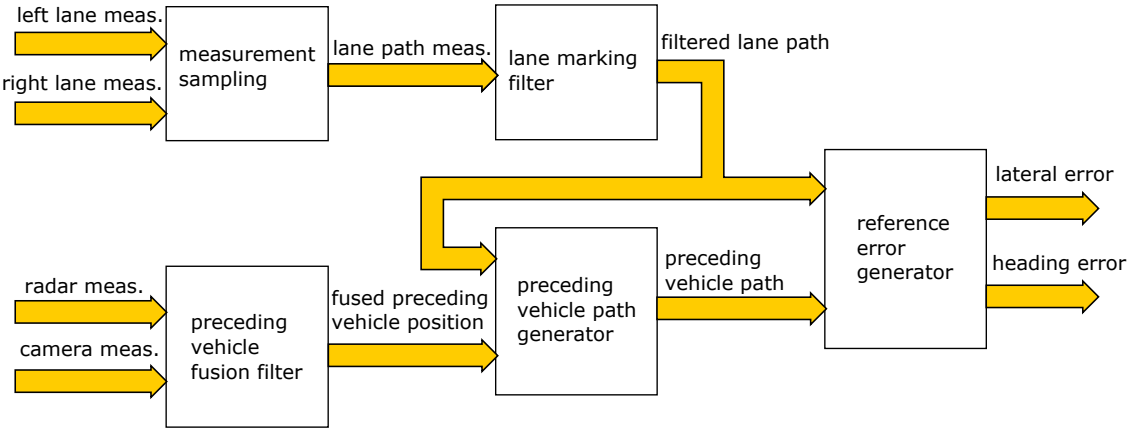


Figure 5.1: Block diagram of the Simulink implementation of the algorithm. A detailed picture of the Simulink implementation can be found the appendix.

measurement sampling

The inputs to this block are the coefficient measurements $\Delta y, \alpha, c_0$ and c_1 (see equation (4.2) page 17) from the left and the right lane marking measurements. The block generates a sampled lane marking path from the measurements as described in sections 4.1.1 and 4.3.2. The output is a sampled lane marking path together with a confidence factor between 0 and 10.

lane marking filter

This block contains the sampled lane marking path filter that is described in section 4.3. The inputs are a sampled lane marking path measurement, a confidence factor between 0 and 10 and the velocity and yaw rate measurements from the proprioceptive sensors. The output is the filtered lane marking path together with its covariance matrix P.

preceding vehicle fusion
: This block contains the filter that fuses the preceding vehicle positions from the radar and the camera explained in section 4.5. The inputs are the lateral and longitudinal position of the preceding vehicle in the Vcs from the radar and camera as well as the lateral and longitudinal velocity from the radar. The output is the filtered position of the preceding vehicle.

preceding vehicle path generator
: This block generates the preceding vehicle path using the preceding vehicle position and measurements from the proprioceptive sensors (see section 4.4). The inputs are the filtered preceding vehicle position, the lane marking path and the velocity and yaw rate measurements from the proprioceptive sensors. The output is the preceding vehicle path.

reference error generator
: This block uses the path generated from the lane markings and the path generated from the preceding vehicle to calculate a lateral error and heading error signal for the lateral controller.

6
Results

6.1 Lane Marking Error Statistics

This section presents a statistical analysis of the lane marking measurements. The left and the right lane marking measurements are analyzed separately. The data set used for this analysis is described in section 4.8. Since there was no dataset available that included lane marking measurements and ground truth, the ground truth was generated from the lane marking measurements combined with Gps data.

6.1.1 Absolute Error

The sensor measurements are evaluated to determine the standard deviations that are used in the measurement covariance matrix in the implemented filter (see section 4.3. The measurement setup is described in detail in section 4.8. Figure 6.1 shows the absolute error between the lane marking measurements and the ground truth at different look ahead distances ($x = \{0, 10m, 20m, 30m, 40m\}$) as a function of time. It can be seen that the error has a time dependent bias which is noticeable at all look ahead distances except for 0m. The error and the time dependent bias are very similar for both the right and left lane marking. The third plot in figure 6.1 shows the curvature of the road corresponding to the measured lane markings. There is no visible correlation between the curvature and the error or the time dependent bias. Other possible causes for this bias could be the pitch or roll of the trucks cabin where the camera is mounted. Such correlations are not investigated in this project, but it can be subject of future work (see Chapter 8).

6.1.2 Error Distribution

A statistical analysis of the error shows that the errors at the evaluated look ahead distances are zero mean and normally distributed. Figure 6.2 and figure 6.3 show the error distributions for the left lane marking measurements and right lane marking respectively. It should be noted that the axis for the histogram for the error at 0m (left most plot in figures 6.2 and 6.3) are different to the axis of the other histograms. The distribution for this error is much narrower than the error distributions at further look ahead distances. This can have two causes. Firstly the lane marking measurement at 0m only depends on the Δy coefficient in the polynomial described in equation (3.2) on page 12 whereas the other measurements are calculated with all the coefficients. Secondly the measurements at 0m were used together with Gps data to generate the ground truth which is described in section 4.8.

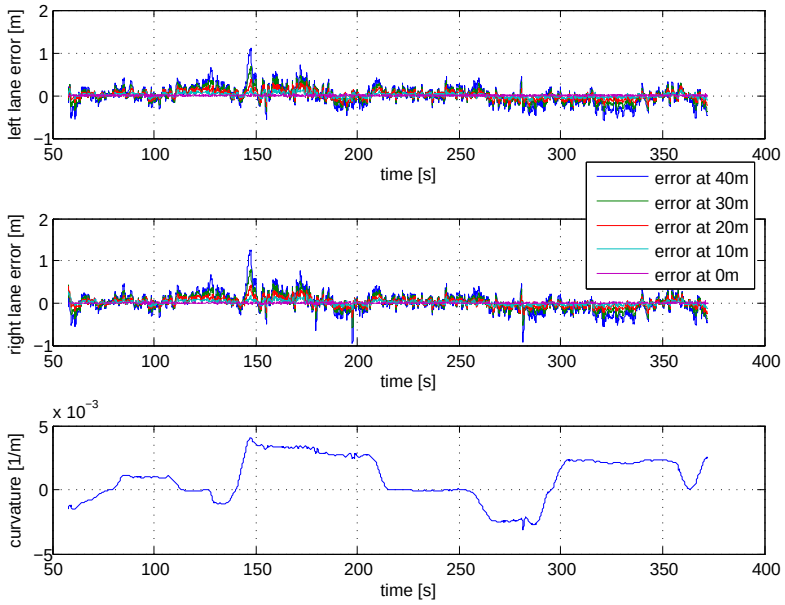


Figure 6.1: Absolute error between lane marking measurements and ground truth at different look ahead distances for the left (top plot) and right (middle plot) lane marking during one measurement run. The curvature of the road (bottom plot) as reference to evaluate a correlation between road curvature and measurement error.

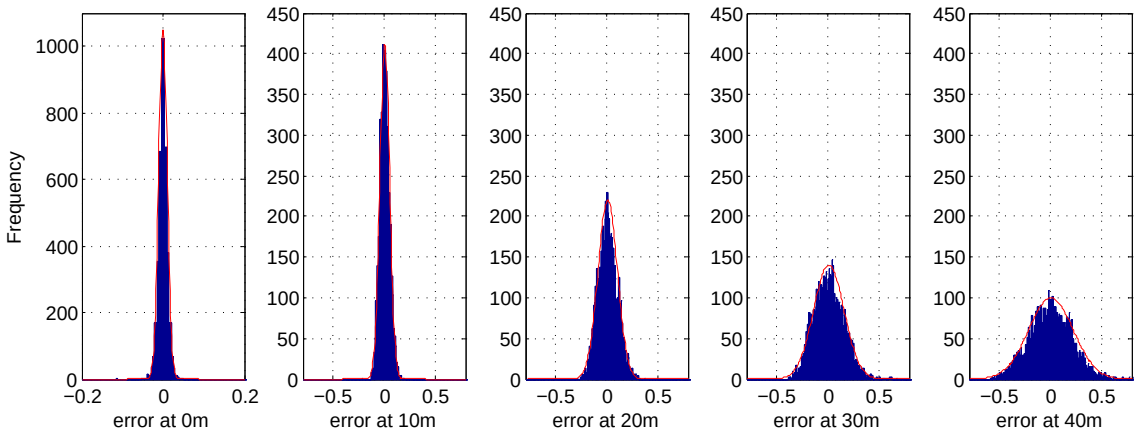


Figure 6.2: Distribution of left lane marking measurement error. All the sets are normally distributed, the red curves are the Gaussian probability density functions with mean values and standard distributions found in table 6.1.

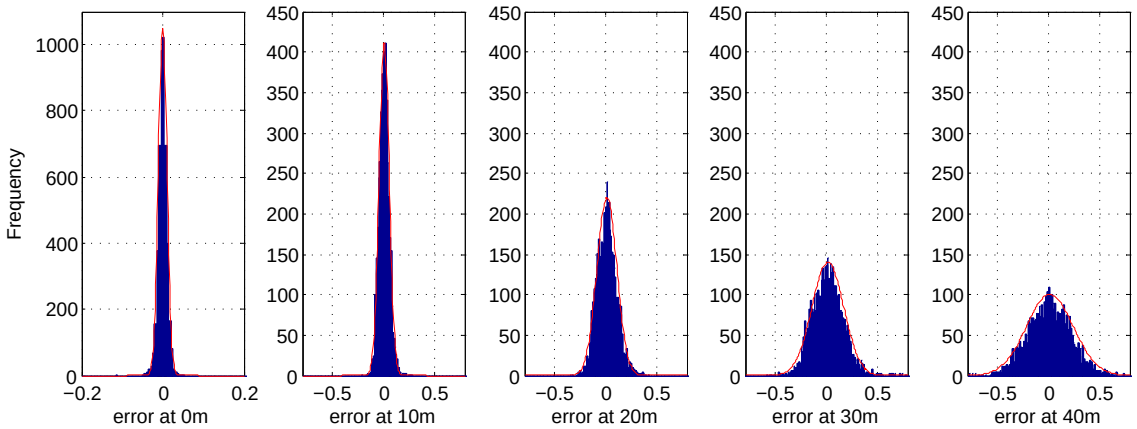


Figure 6.3: Distribution of right lane marking measurement error. All the sets are normally distributed, the red curves are the Gaussian probability density functions with mean values and standard distributions found in table 6.1.

Dist:	0m	10m	20m	30m	40m
error distribution left lane marking					
mean [m]:	0	0.0068	0.0105	0.0105	0.0057
std. Dev. [m]:	0.0103	0.0473	0.0949	0.1501	0.2197
error distribution right lanemarking					
mean[m]:	-0.0001	0.0075	0.0125	0.0146	0.0128
std. Dev.[m]:	0.0104	0.0491	0.0981	0.1566	0.2309
error distribution combined					
mean[m]:	0	0.0072	0.0115	0.0126	0.0093
std. Dev.[m]:	0.0104	0.0482	0.0965	0.1534	0.2254

Table 6.1: Distributions of the lane marking measurement errors. Left and right lane separate and both lanes combined.

6.2 Lane Marking Filter Comparison

In Schwartz paper [14], he states that filtering the coefficients is very sensitive to noise since the noise in one coefficient might propagate to other coefficients as they are dependent on each other. He states that a range and error filter as the sampled method explained in section 4.3 is more suitable for filtering. Both filters were implemented and compared although a statistical comparison was not made. It was clear from visualization that the sampled filter was superior to the coefficient filter. In figure 6.4 a comparison of the the prediction steps can be seen. This is done by simulating that the measurements are not available at certain interval and observing how close the prediction is to the "hidden" measurements. In figure 6.4 a) the measurement is available and the update step is executed but In figure 6.4 b),

c) and d) only the prediction step is executed. It can be seen from the plots that the coefficient filter moves away from the measurements while the sampled filter performs quite well. In figure 6.4 d) when the vehicle has driven one second (about 17 meters) without measurement the error is within 10 cm for the sampled filter but over 40 cm for the coefficient filter.

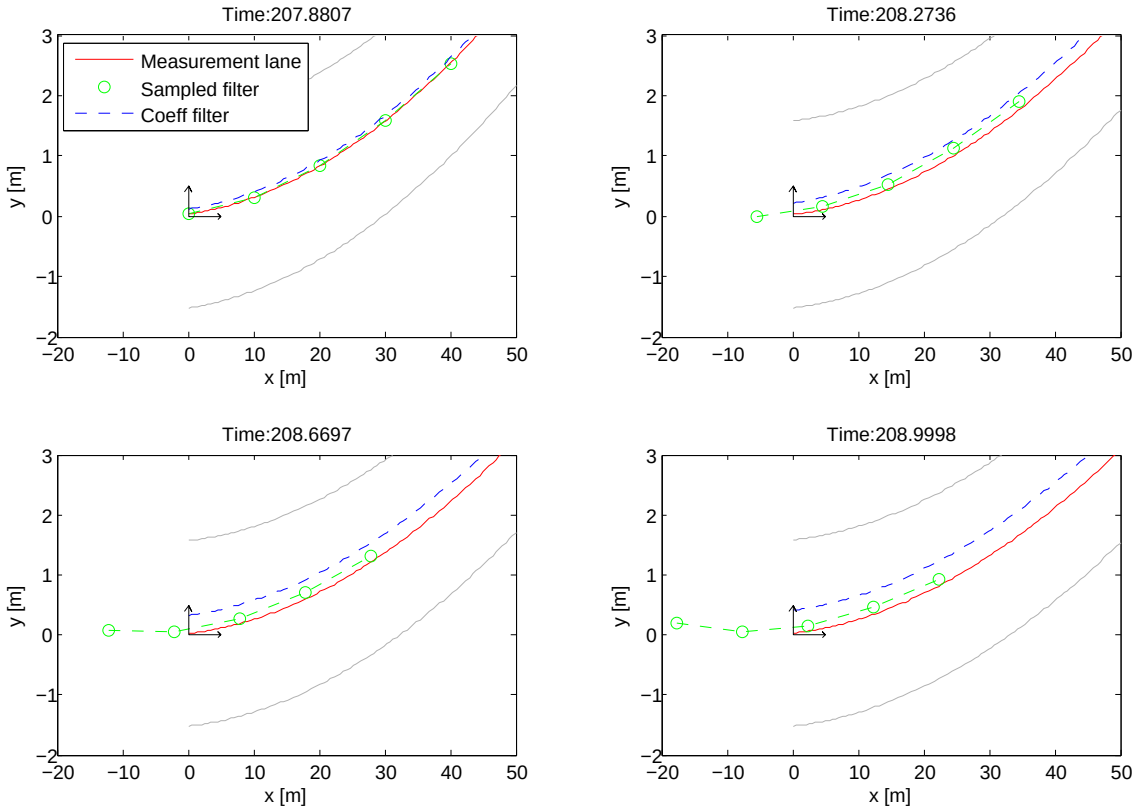


Figure 6.4: The figures illustrate the performance of the prediction step in sample filter and coefficient filter. The gray lanes are the actual measurements, red line is the measurements of left and right lanes averaged, green circles are the estimates from the sampled filter and the blue line is the estimate from the coefficient filter.

6.3 Lane Marking Filter

In this section the performance of the sampled lane marking filter described in section 4.3 is presented. The results are based on the same dataset that is used for the analysis of the sensor performance in section 6.1. It was collected under very good measurement conditions at AstaZero. An overview of the track that the vehicle drove while collecting the data set can be found in figure 4.9. In the following sections the performance of the filter is compared to the unprocessed measurements. First the filtered data is compared to the measurements as they were collected, then the performance is compared when noise is added to the measurements and finally the performance is compared when measurements are removed from the dataset. In section 6.3.1 the error of all the points are compared to the measurements and in sections 6.3.2 and 6.3.3 the lateral control signal is compared. The lateral control

signal (see section 4.6) is the lateral deviation at a look ahead distance of 1s (see section 4.6). This means that the look ahead distance depends on the velocity of the vehicle.

6.3.1 Performance with original sensor data

Here the filter is compared to the measurements using the unaltered dataset. Figure 6.5 shows the error of the filtered lane marking path at different look ahead distances ($x = 0m, 10m, 20m, 30m, 40m$). It can be seen that the error distribution is very similar to the distribution for the unprocessed measurements seen in figure 6.6. The error signals are overall a bit smoother as can be seen in the zoomed sections of figures 6.5 and 6.6. The reason these two signals are so similar has two explanations. Firstly the measurements were collected in very good conditions for lane marking camera and have therefore little noise. Secondly the main contribution to the error is a time dependent bias that cannot be reduced by filtering.

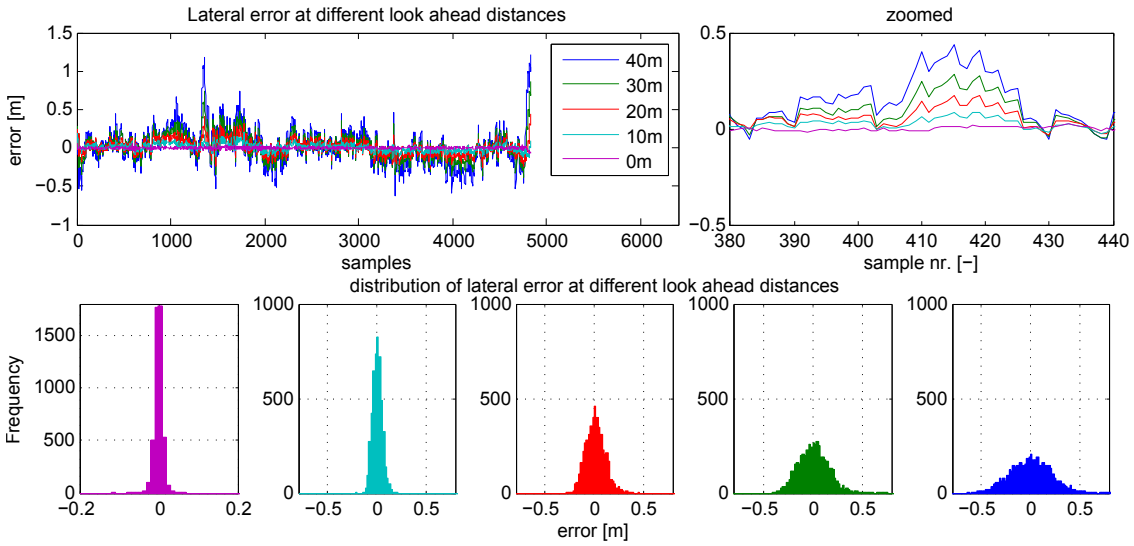


Figure 6.5: Error to ground truth for the unprocessed lane marking path. (top left) error at look ahead distances $x = 0m, 10m, 20m, 30m, 40m$, (top right) zoomed part of error, (bottom plots) histograms for the distribution of the errors with colors corresponding to signals in the top plots.

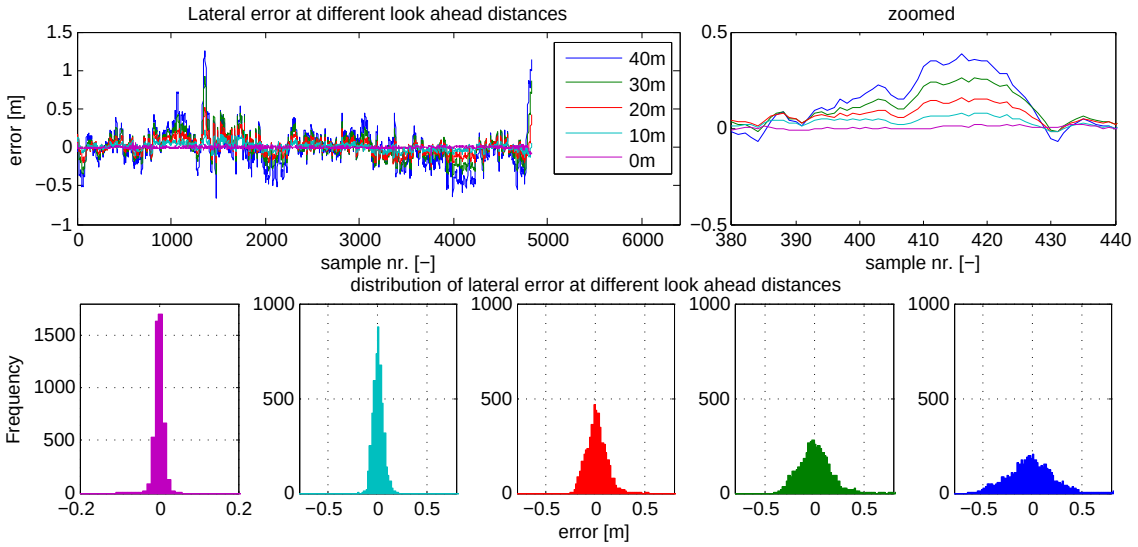


Figure 6.6: Error to ground truth for the filtered lane marking path. (top left) error at look ahead distances $x = 0m, 10m, 20m, 30m, 40m$, (top right) zoomed part of error, (bottom plots) histograms for the distribution of the errors with colors corresponding to signals in the top plots.

The lateral control signal that is sent to the controller can be seen in figure 6.7. It both shows the control signal calculated from the filtered path and from the unprocessed measurements. The error is the difference between the lateral control signal and the control signal calculated from the ground truth. It also shows the probability density function of a fitted normal distribution for the error from the measurements and the error from the filtered path. Also in this analysis the filtered signal and the unprocessed measurements are very similar.

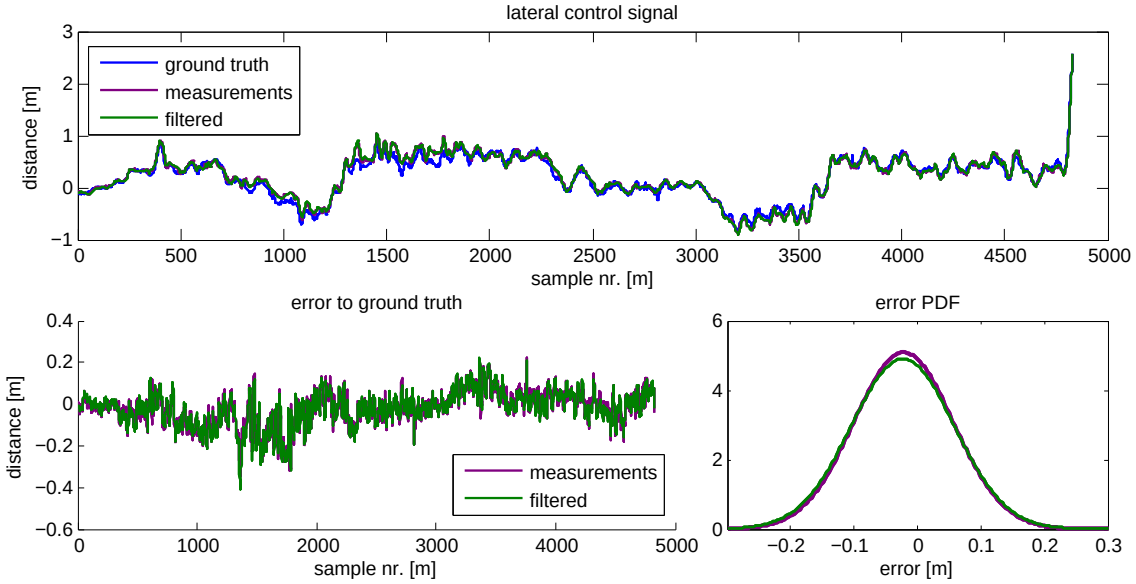


Figure 6.7: (top) lateral control signal generated from ground truth, unprocessed lane path measurements and filtered lane marking path, (bottom left) error to ground truth signal, (bottom right) probability density functions for errors.

6.3.2 Filter Performance With Added Noise

In this test noise is added to all the measurements. The noise is added to all the coefficients of the lane marking measurements for the left and right lane ($\Delta y, \alpha, c_0, c_1$ see equation (4.2)). The added noise is white with a Gaussian distribution. Figure 6.8 and figure 6.9 show the error on the lateral control signal from the filter estimates and the unprocessed measurements with added noise of 10% and 20% of the individual coefficients respectively. Table 6.2 shows the mean values and standard deviations for the error on the unprocessed measurements and the filter estimates for the unaltered dataset and the datasets with added noise. This shows that the implemented filter is robust towards measurement noise.

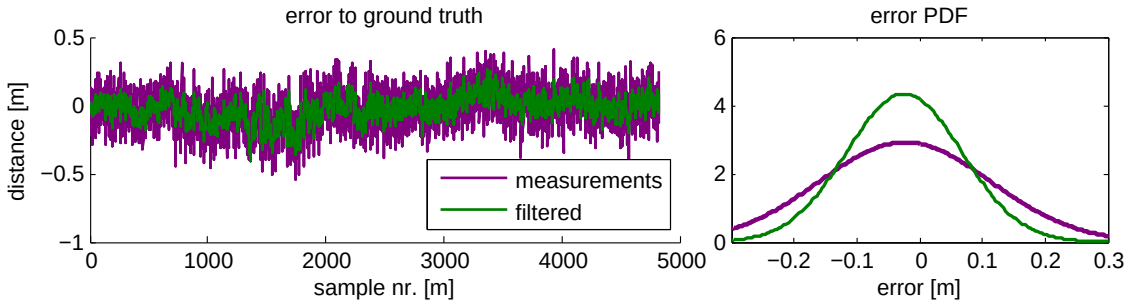


Figure 6.8: Error of the lateral control signal to ground truth for unprocessed measurement path and filtered path with added 10% noise to the measurements.

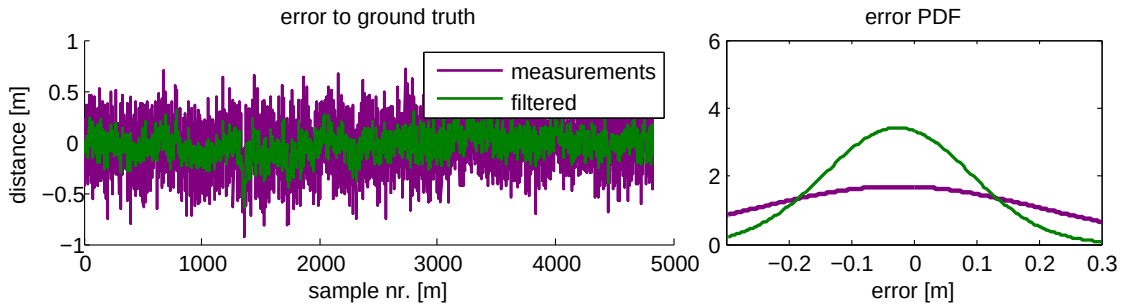


Figure 6.9: Error of the lateral control signal to ground truth for unprocessed measurement path and filtered path with added 20% noise to the measurements.

	mean value [m]	std. deviation [m]
unaltered dataset		
unprocessed measurements:	-0.0220	0.0784
filtered measurements:	-0.0232	0.0814
dataset with 10 % added noise		
unprocessed measurements:	-0.0226	0.1380
filtered measurements:	-0.0240	0.0929
dataset with 20 % added noise		
unprocessed measurements:	-0.0179	0.2341
filtered measurements:	-0.0192	0.1138

Table 6.2: Mean values and standard deviations of the errors for the datasets with added noise.

6.3.3 Filter Performance Removed Measurements

For this test measurements were randomly removed from the input dataset to simulate the the sensor not detecting the lanes. Figures 6.10 and 6.11 show the result for when 20% of all measurements are removed from the dataset. For the analysis in figure 6.10 the lateral control signal of the unprocessed measurements is set to 0 when the measurements are removed. This is because the sensor outputs are 0 when no lanes are detected.

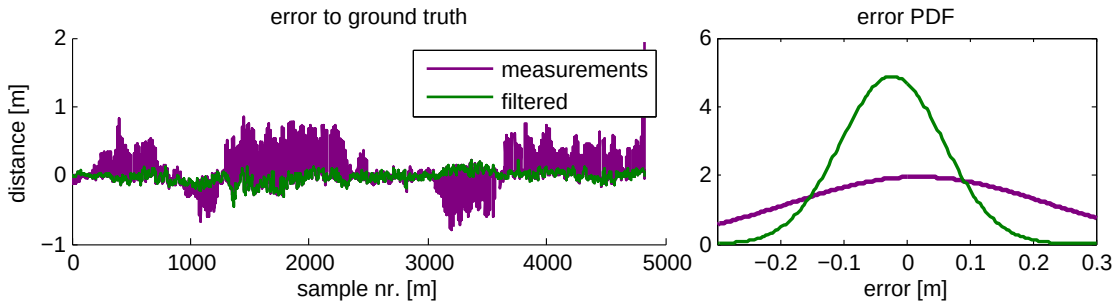


Figure 6.10: Error of the lateral control signal to ground truth for unprocessed measurement path and filtered path with 20 % of measurements removed from the path. When no measurement is available the lateral control signal for the unprocessed measurement path is set to 0.

In the analysis in figure 6.11 the lateral control signal is kept constant when no lane measurements are detected. For this dataset it seems as if keeping the lateral control signal constant is equally good or even slightly better than using the filtered signals. This is because in the dataset that is used for this test the driver kept the vehicle very well in the center of the lane which makes the lateral control signal constant during a curve. Using this strategy in a real world implementation might prove to be dangerous as the system has no information about the position of the vehicle in the lane once the measurements are lost.

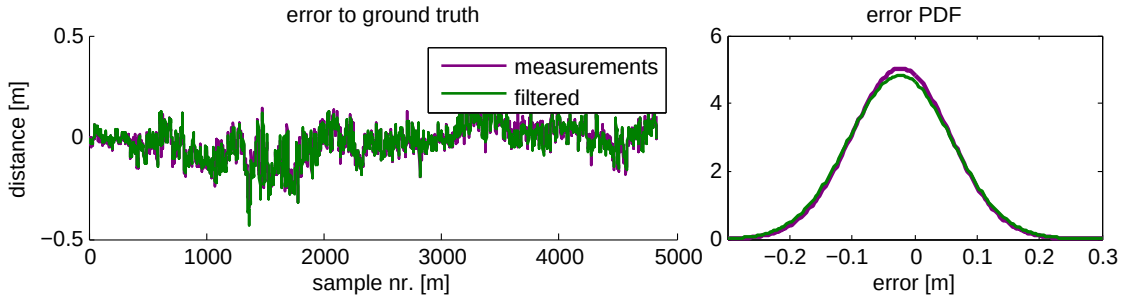


Figure 6.11: Error of the lateral control signal to ground truth for unprocessed measurement path and filtered path with 20 % of measurements removed from the data set. When no measurement is available the lateral control signal for the unprocessed measurement path is kept constant at the last value.

Figures 6.12 and 6.13 show the analysis for when 85% of the measurements are removed from the data set. In figure 6.12 the error signal is kept constant when in figure 6.13 the error signal is set to 0. A similar behavior can be seen as when 20% of the measurements were lost. The filter is still able to estimate the path quite well. A closed loop test would be needed in order to see if keeping the reference as constant would work as well. Table 6.3 shows the mean values and standard deviations of the error signals for the tests when the data set is unaltered, 20% of the measurements are lost and 80% of the measurements lost.

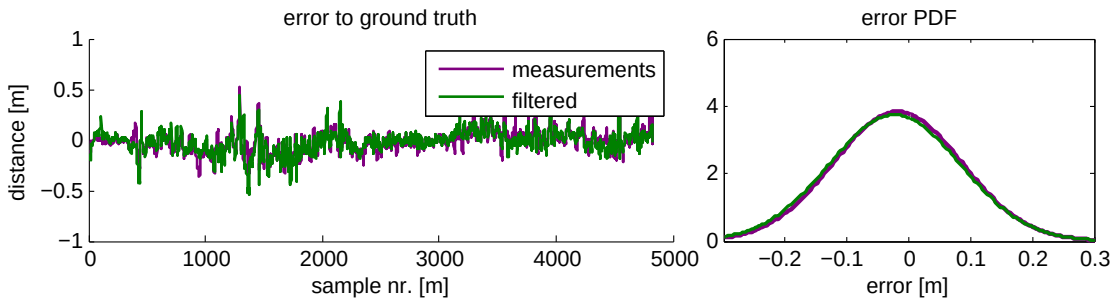


Figure 6.12: Error of the lateral control signal to ground truth for unprocessed measurement path and filtered path with 85 % of measurements removed from the data set. When no measurement is available the lateral control signal for the unprocessed measurement path is kept constant at the last value.

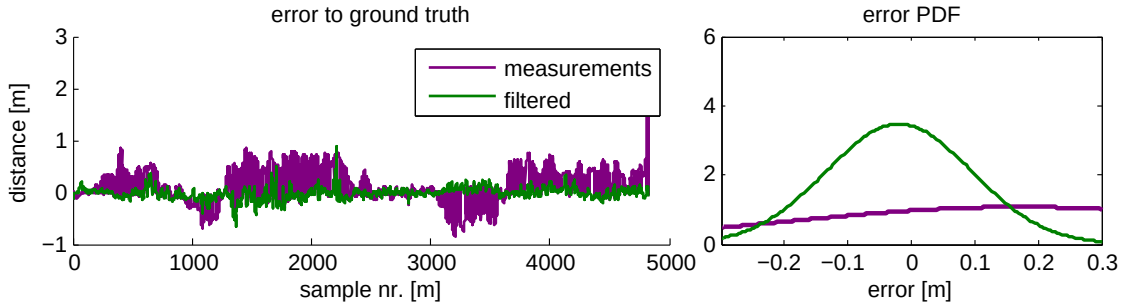


Figure 6.13: Error of the lateral control signal to ground truth for unprocessed measurement path and filtered path with 85 % of measurements removed from the data set. When no measurement is available the lateral control signal for the unprocessed measurement path is set to 0.

6.4 Preceding vehicle filter analysis

The measurements of the object detection from radar and camera sensors are sampled asynchronous. In section 4.9 a description of how the measurements were filtered and smoothed to generate ground truth can be found. The ground truth is used as reference in order to determine the step length of the filter and if the measurement are to be treated as asynchronous or not. In table 6.4 the mean square error between the filter estimates and the ground truth can be seen using different step lengths (dt). Fixed dt means that the distribution is estimated with fixed interval using the latest measurements that have been sampled if any since the last update step. With varied dt the distribution is found at the correct time instance when the measurement is sampled but some measurements might be skipped if the dt is larger than the sampling frequency. Treating the data as asynchronous does produce less error than treating them synchronous which is expected but the difference is quite small. The small difference can be explained by the fact that the Ego vehicle and the preceding vehicle travel a similar path at similar velocities and

	mean value [m]	std. deviation [m]
unaltered dataset		
unprocessed measurement path:	-0.0220	0.0784
filtered lane marking path:	-0.0232	0.0814
dataset with 20 % removed measurements		
unprocessed measurement path:	0.0270	0.2071
filtered lane marking path:	-0.0229	0.0818
meas. with lateral control kept const.:	-0.0219	0.0789
dataset with 85 % removed measurements		
unprocessed measurements:	0.1681	0.3663
filtered measurements:	-0.0174	0.1225
meas. with lateral control kept const.:	-0.0207	0.0968

Table 6.3: Mean values and standard deviations of the errors for the data sets with removed samples. comparison between filtered lane marking path, unprocessed measurement path with control signal set to 0 when measurement is removed and unprocessed measurement path with control signal kept constant when measurement is removed.

Mean square error		
dt [s]	**Fixed dt**	**Varied dt**
0.01	0.0027	0.0023
0.05	0.0040	0.0026
0.1	0.0043	0.0033
0.2	0.0044	0.0034
0.5	0.0046	0.0035

Table 6.4: Mean square error of the lateral position. The error is between the ground truth and filter estimates with different time step sizes (dt).

therefore the change in measurements between time steps is small. When the path is generated from the preceding vehicle the points in the path move according to dead reckoning as explained in section 4.4.1. This produces uncertainty that is much larger than the uncertainties that come from treating the data as synchronous.

6.5 Preceding vehicle path

A reference path was constructed from the filtered preceding vehicle position according to section 4.4.1. The coordinate transformation of the points that represent the path in Ego's coordinate system is based on dead reckoning of the Ego vehicle. This results in increasing uncertainty as the points move away from the preceding vehicle. The path from the preceding vehicle can be seen in figure 6.14. The path from the preceding vehicle is not as smooth as the lane path since the preceding vehicle does not always drive in the middle of the road. It might be of interest to smoothen

the path before calculating the control signals but that was not investigated further in this thesis.

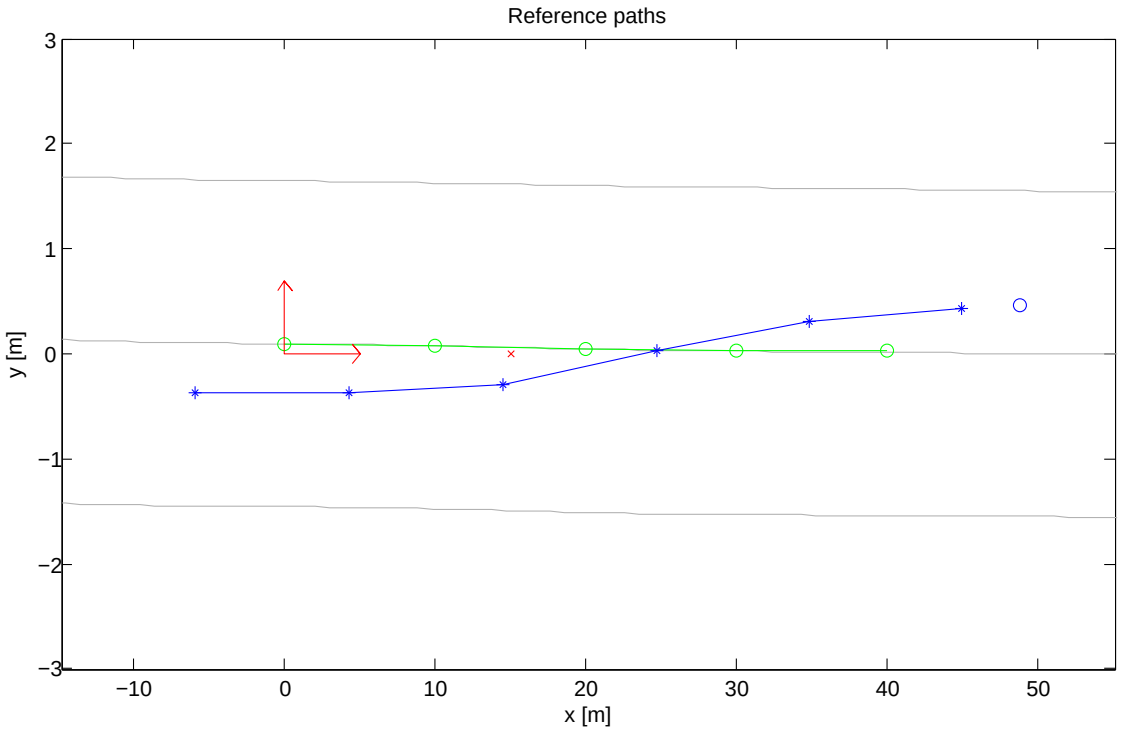


Figure 6.14: The paths that are generated for the lateral control. The green path is the lane marking reference path, blue line is the preceding vehicle path, blue circle is the current estimated position of the preceding vehicle, red cross is the look ahead point and the gray lanes are the right and left lane markings.

6.6 Outlier rejection

When the preceding vehicle goes out of sensors field of view the sensor outputs a constant value. In figure 6.15 the measurements from the radar and camera can be seen. As soon as the radar loses the object it outputs the measurement 102 with the sigma value 0. The filter would trust this measurement as the uncertainty is zero and set the estimate of the lateral distance close to 102 meters. By using outlier rejection which rejects the measurement if it is over 100 this behavior will be accounted for.

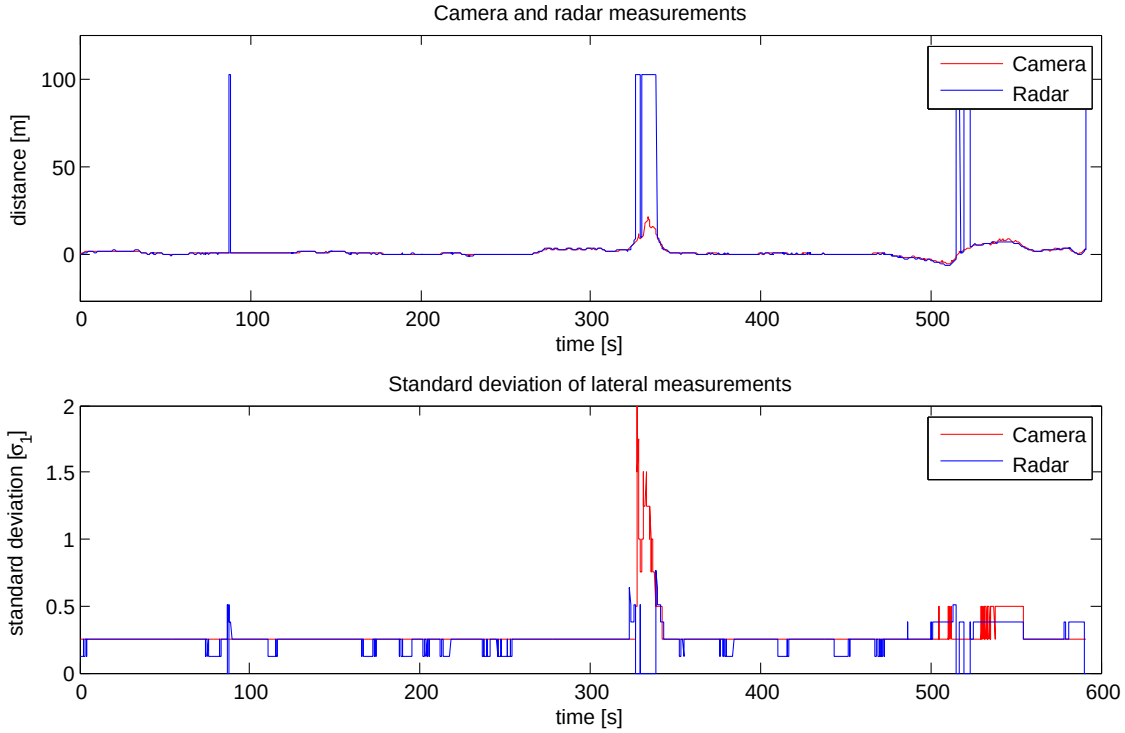


Figure 6.15: The radar and camera measurement of lateral position of the preceding vehicle. The standard deviation for the measurements that is estimated by the sensor can be seen in the lower plot.

7

Conclusion and Discussion

By using the implemented filters in this thesis two reference paths were generated for lateral control. The filters use sensor information from camera and radar to achieve robust lateral reference signal. This method is superior from using a raw measurement from a single source as a reference since it can handle noisy measurements and rely on prediction when measurements are lost. In order to have a reference path a lane markings or preceding vehicle need to be detected by the sensors. The scope of this thesis was to use sensors that are currently available in production vehicles at Volvo Group Trucks Technology. State of the art sensors that are more reliable and accurate were discarded since they are not yet feasible for commercially available vehicles.

The solution proposed in this thesis is supposed to be used as a base platform for how sensor data are handled in several projects that are currently in process at the Atr department.

7.1 Implemented filters

The filters were implemented for estimation of lane marking and position of preceding vehicle. Two filters were implemented and compared for the lane markings. The filter based on range and error (sampled) proved to perform much better than the coefficient filter as seen in section 6.2 so it was further developed. In the sampled lane marking filter the uncertainties of the camera measurements were quantified using methods described in section 4.8. The resulting uncertainties describe the distribution of the measurements and are important to accurately set the measurement covariance in the filter. The filter is able to handle measurements that are affected by large noises as can be seen in section 6.3.2. The main advantages of the lane marking filter is that if the measurements are lost for short period of time the filter can still provide reference by using prediction of the path for up to 40 meters. This can be beneficial in scenarios where lane marking is partly obscured by e.g. snow or dirt. The second filter uses the object detection from the camera and radar sensors and estimates the position and velocity relative to the Ego vehicle. Although the outputs from the sensors are already processed data we treat them as raw data since we do not have access to the raw ones.

7.2 Reference paths and error signals

Two paths were generated for lateral reference. One path from the lane marking estimates and another from the estimated position of the preceding vehicle. Each path has a specific uncertainty throughout the path. The uncertainty in the lane path is usually much smaller than in the preceding vehicle path. That is mainly caused by the fact that we have measurements along the entire lane path but only at the end of the preceding vehicle path. The control signal is calculated from the lane markings whenever that path is available. As soon as the measurements are lost for the lane and uncertainty increases the reference error is transitioned over to the preceding vehicle path.

7.3 Experimental implementation

The algorithm was developed using Matlab but then converted to Simulink model for implementation in rapid prototype environment (dSpace) which is available in the Volvo test trucks. Due to time restrictions only one test was done on the model. The model was able to output reasonable references but was not tested in closed loop with the controller. Further tests are scheduled to take place in the upcoming weeks.

8
Future Work

As our algorithm is a platform in how to handle sensor data for lateral control there are manz possibilities to add to this platform. Here we list those that we consider the most important ones.

Other sensor sources

- **GPS / map:** GPS together with a map could provide useful information. One example is at exits on highways where one lane turns off the highway while the other stays on. This can cause a problem for the lane detection sensor, when it detects one lane marking following the road and the other following the exit road. There the vehicle position on a map could provide information about which lane to use.
- **LIDAR:** A LIDAR sensor provides very accurate view of the surroundings. The information could be used to improve the estimation of the paths.
- **Other camera information:** Such as road edges, measurements of lanes next to the vehicle lane and roadsigns would be useful.
- **Additional cameras:** In this thesis a front view camera is used. Cameras on the side or on the back of the truck could be used to provide information on other vehicles surrounding the truck.
- **V2V communication:** By incorporating vehicle to vehicle (V2V) communication information about other vehicles could be useful. A preceding vehicle with the same system as described in this thesis could communicate the paths estimated in it's vehicle. That path could be used to extend the paths in the Ego vehicle.

State estimator Currently the distance traveled and change in heading is calculated using raw measurements from the speedometer and yaw rate. This could be improved by estimating the state of the vehicle using sensor fusion. This estimator could be used straight into the prediction part of the filters. By using the state estimator a more accurate change in heading and distance traveled between time steps would be achieved.

Initial preceding vehicle path estimation The estimation of the initial preceding vehicle path when no lane marking path is available could be improved. In the proposed solution the path is set by linear interpolation between the Ego

vehicle and the preceding vehicle. Since the heading of the EGO vehicle is known a cubic interpolation could be used to get the estimated path to follow the curve of the road.

Reference paths The reference paths are currently represented with 5 to 6 points which could cause problems for the controller. The paths could be smoothed by drawing a bezier curve through the points before calculating the control signal. This method was looked into but not investigated further because of time constraints on the project.

Time variant bias in lane marking measurements The lane marking measurements has a time variant bias. The bias is not correlated with the curvature of the road as can be seen in section 6.1. The cause of the bias could be related to orientation of the lane marking camera since it is at the top of the windshield where it is affected by yaw and roll of the cabin. The roll and the yaw of the cabin could be measured using accelerometers and compared to the lane marking error. If there is a correlation it can be accounted for and therefor improve the measurements.

Bibliography

[1] E. Chan, P. Gilhead, P. Jelínek, P. Krejčí, and T. Robinson, "Cooperative control of sartre automated platoon vehicles," *ITS World Congress 2012*, 2012.

[2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[3] I. Papadimitriou and M. Tomizuka, "Lateral control of platoons of vehicles on highways: the autonomous following based approach," *International Journal of Vehicle Design, September 27*, vol. 36, no. 1, 2004, pp. 24–37, 2004.

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2005.

[5] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transaction on Automatic Control*, vol. 54, no. 6, 2009.

[6] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, 1965.

[7] J. Kosecka, R. Blasi, C. J. Taylor, and J. Malik, "Vision-based lateral control of vehicles," *IEEE Conference on Intelligent Transportation Systems, 9-12 November, Boston, MA*, pp. 900 – 905, 1997.

[8] J. Kosecka, R. Blasi, C. J. Taylor, and J. Malik, "A comparative study of vision-based lateral control strategies for autonomous highway driving," *1998 IEEE International Conference on Robotics and Automation 20-20 May, Leuven, Belgium*, vol. 3, pp. 1903 – 1908, 1998.

[9] M. Tai and M. Tomizuka, "Experimental study of lateral control of heavy vehicles for automated highway systems (ahs)," *Proceedings of the American Control Conference, 8-10 May, Anchorage, AK*, pp. 851 – 856, 2002.

[10] J. Huang and H.-S. Tan, "Immediate transition for an automated bus using magnetic and dgps sensing systems," *14th International IEEE Conference on Intelligent Transportation Systems, 5-7 October, Washington, DC, USA*, pp. 656 – 661, 2011.

[11] Y. Hayakawa, R. White, T. Kimura, and G. Naitou, "Driver oriented path following in its - wide speed-range steering control by multiple look-ahead distances," *Proceedings of IEEE/ASME International Conference on Advanced Intelligen Mechatronics, 20-24 July*, vol. 1, pp. 558 – 563, 2003.

[12] M. Tai and M. Tomizuka, "Feedforward compensation for lateral control of heavy vehicles for automated highway systam (ahs)," *IFAC 15th Triennial World Congress, Barcelona, Spain*, 2002.

[13] K. Hasegawa and E. Konaka, "Three look-ahead distance scheme for lateral control of vision-based vehicles," *SICE Annual Conference, September 9-12, Hakkaio University, Sapporo, Japan*, pp. 660 – 665, 2014.

[14] D. Schwartz, "Clothoid road geometry unsuitable for sensor fusion," *IEEE Intell. Veh. Symp*, pp. 484–488, 2003.

[15] Ángel F. García-Fernández, L. Hammarstrand, M. Fatemi, and L. Svensson, "Bayesian road estimation using onboard sensors," *IEEE Trans. On ITS*, vol. 15, 2014.

A
Appendix 1

A.1 Maps

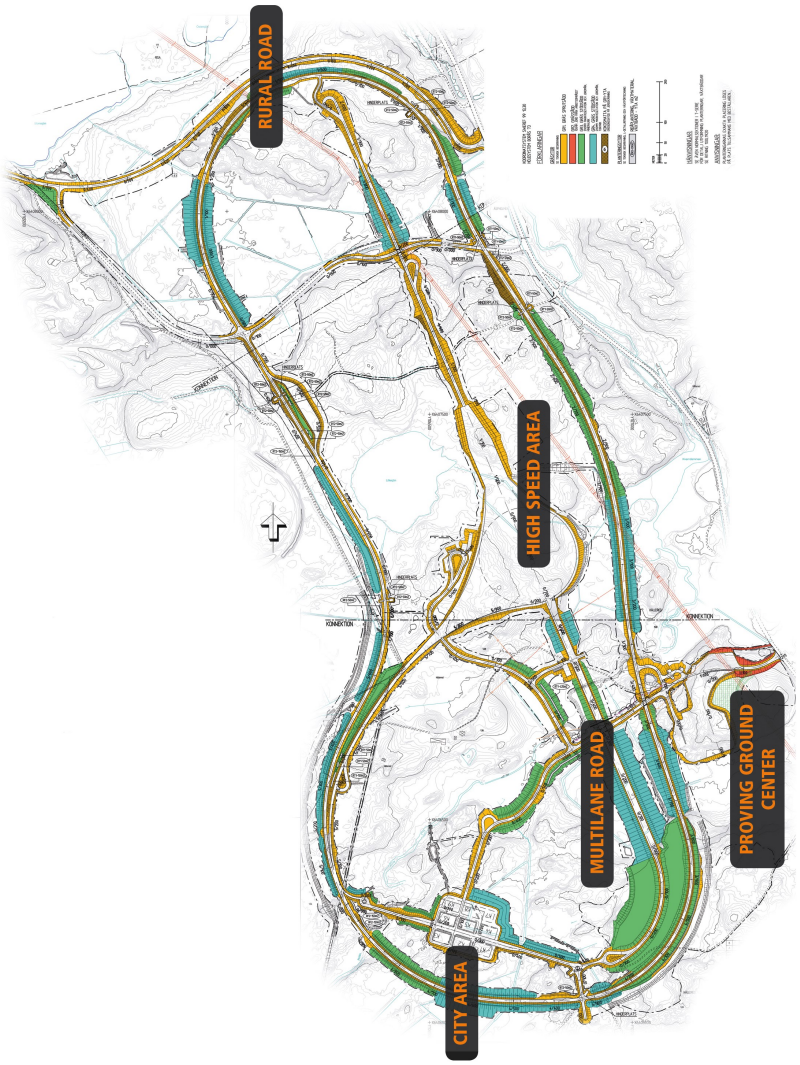


Figure A.1: Detailed map of the testing area AstaZero
II

A.2 Simulink

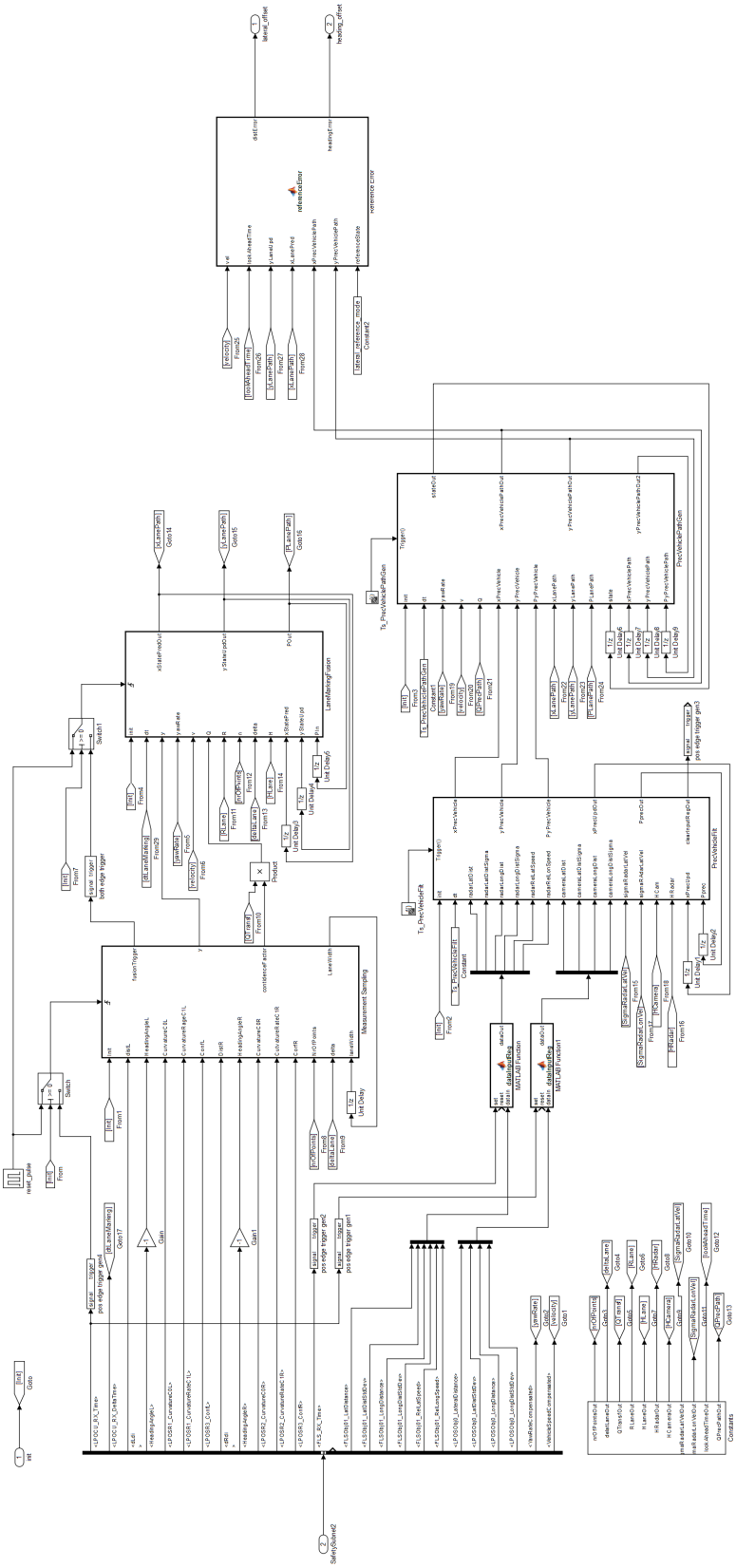


Figure A.2: Simulink implementation of the algorithm proposed in this thesis