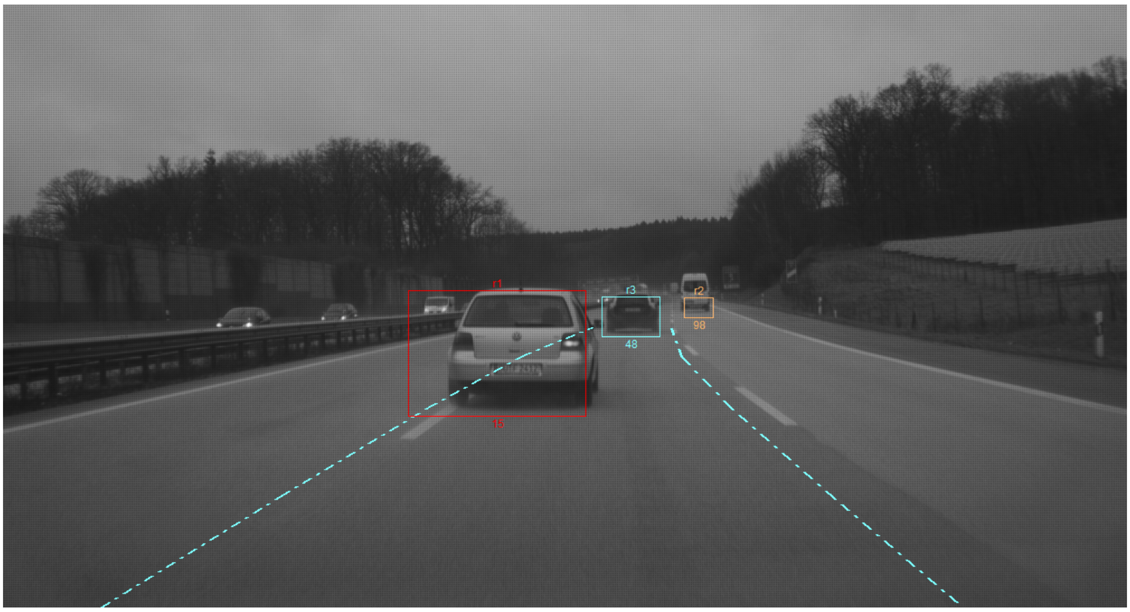
CHALMERS
UNIVERSITY OF TECHNOLOGY



Road geometry estimation for longitudinal vehicle control

A method of using stationary detections for road geometry estimation

Master's thesis in Systems, Control and Mechatronics

ADAM PETTERSSON
VIKTOR SVENSSON

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

Master's thesis EX076/2015

Road geometry estimation for longitudinal vehicle control

A method of using stationary detections for road geometry estimation

ADAM PETTERSSON
VIKTOR SVENSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

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

Road geometry estimation for longitudinal vehicle control
A method of using stationary detections for road geometry estimation
ADAM PETTERSSON
VIKTOR SVENSSON

Supervisors: Lars Hammarstrand, Department of Signals and Systems
Kenny Karlsson, Delphi Automotive

Examiner: Lennart Svensson, Department of Signals and Systems

Master's Thesis EX076/2015
Department of Signals and Systems
Signal Processing Group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2015

Road geometry estimation for longitudinal vehicle control
A method of using stationary detections for road geometry estimation
ADAM PETTERSSON
VIKTOR SVENSSON
Department of Signals and Systems
Chalmers University of Technology

Abstract

The presence of Advanced Driver Support Systems (ADAS) in vehicles on the roads today is constantly increasing. These types of systems don't only make the vehicles safer, they also increase the comfort of the driver. The Adaptive Cruise Control (ACC) function consists of a number of steps, of which the road modelling and the threat assessment are two of the most important parts. To do a reliable threat assessment, an accurate model of the road ahead is necessary. The process of creating a road model takes into account multiple sources of information, such as lane marker detections and guardrail estimations.

This thesis investigates the potential of using stationary targets detected by the sensor system to enhance the predicted road model. The aim is to derive and implement an algorithm to estimate the road geometry by utilizing the information of lane marker detections, guardrail estimations, preceding vehicles and single stationary objects. The implemented method uses a Bayesian filter which has curvature sampled at fixed distances as states. The Extended Kalman Filter (EKF) is used for the non-linear models.

The result of the thesis shows that using all available information sources greatly improves the road geometry estimation, compared to only using the lane marking detections as an estimate of the road ahead. The addition of using single stationary object detections is shown to have little or no effect on the Root-Mean-Square Error (RMSE) of the estimated road. Non obstructing stationary objects, for example overhead road signs or bridges, are being detected on the road which are causing the estimated road geometry to deviate. Without more information about the stationary objects, such as the elevation, only using the stationary object detections to create an accurate guardrail estimate is concluded to be a more useful method for the road geometry estimation problem.

Keywords: Advanced Driver Support Systems, Road geometry, Adaptive Cruise Control, Bayesian filter, Extended Kalman Filter

Acknowledgements

We would like to thank our supervisor at Delphi, Kenny Karlsson, for his constant support and guidance. We would also like to thank Delphi for providing the necessary hardware, such as test vehicles, to make this thesis possible.

Special thanks go to our supervisor, Lars Hammarstand, and our examiner, Lennart Svensson, at Chalmers Univesity of Technology for sharing your knowledge and always keeping us on the right track.

Adam Pettersson & Viktor Svensson, Gothenburg, December 2015

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

Advanced Driver Assistance Systems (ADAS), are becoming increasingly common in vehicles seen on our roads. Functions as Automatic Emergency Braking (AEB) and Blind Spot Detection (BSD) have the potential to reduce the number of accidents and systems like Adaptive Cruise Control (ACC), Traffic Jam Assist (TJA) and future highway pilot functions can reduce the stress level of drivers and increase safety. In this way, the driving experience becomes more relaxed and less tired, leading to more observant drivers.

The early ADAS systems have been used on the roads for several years and the statistics show that the systems are effective. In a European research project, performed in 2012, it is shown that even the convenience functions, such as ACC, can reduce the number of critical situations on a motorway scenario with as much as 70% [1]. The Insurance Institute for Highway Safety (IIHS), which is a non profit U.S insurance organisation, have compared insurance claims from owners of a Volvo XC60 to other mid size luxury SUVs with data from the years 2009-10. The result shows that the Volvo equipped with a version of city safety has a 27% lower property damage liability rate, 51% lower bodily injury liability and a 22% lower collision rate compared to the other SUVs [2]. These systems have since improved and are active in a wider range of scenarios.

Delphi Corporation is a diverse company, developing and manufacturing components to different automotive manufacturers and for different purposes. A department of Delphi is developing sensors and software used for ADAS. Electronically Scanning Radar (ESR) and Intelligent Forward View (IFV) cameras are two products in the company's portfolio. These products enables features like AEB, ACC and Automatic Headlight Control (AHC).

The requirements from customers are increasing and requests that the supplier should handle more of the complete function, not only the sensing part, but also the target selection and controller design, are more frequent. Therefore suppliers, like Delphi, need to cover a larger scope with their products.

The ACC function, also called *longitudinal vehicle control*, consists of a number of steps, from sensing and fusion of data to target selection and vehicle control. The step of choosing a target includes several sub-functions. Two of the most important parts are the road modelling and the threat assessment.

In order for the function to do an accurate threat assessment, a good model of the road ahead is necessary. Especially the lateral position of the road at different longitudinal positions is important information for the threat assessment step. To be able to determine which lane a certain target vehicle in-front is placed, the position of the road ahead is critical information. If the position estimate from the road modelling step is more than a lane width wrong, the target selection step will not be able to do a correct lane assignment.

The process of creating a road model takes into account multiple sources of information. The sources include lane marker information, movement of other vehicles and estimations of the guardrails. There are several ways of combining these information sources and research is being performed within the area [3].

This thesis investigate the possible methods and potentials of using stationary targets detected by the sensor system to enhance the predicted road model. There exist methods to use stationary targets to estimate guardrails and barriers, but few or no methods to take single stationary targets, such as posts or road signs, into account when estimating a road model. The positions where such targets are located are likely to be outside of the road, providing information to the road modelling about where the road is likely not to be.

1.1 Problem formulation

The problem in this project is to estimate the shape of the road ahead of the vehicle equipped with a radar and a camera sensor. The shape is supposed to be estimated at a distance up to 200 meters in front of the vehicle and the estimate should be focused on performance needed for longitudinal vehicle control. Figure 1.1 is showing an example scenario and the geometry to be estimated in this case. The blue vehicle symbolize the host vehicle and the red dotted line is the road shape which is supposed to be estimated.

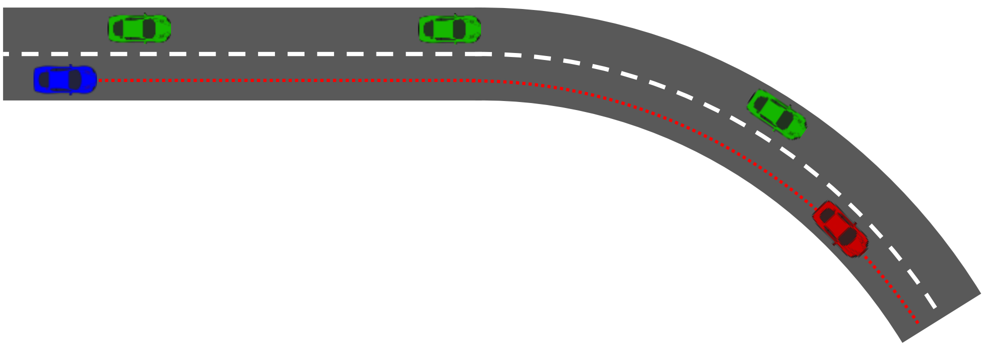


Figure 1.1: Illustration of the geometry to be estimated.

As mentioned in the introduction, a road model can be created using multiple sources of information. Typically the following sources are used:

1. Current motion of host vehicle, e.g. speed, yaw rate, slip angle etc.
2. Observation of lane markings
3. Estimation of guardrails or barriers
4. Properties of vehicles ahead

There is, however, situations when this information is not enough to make an accurate road model. Examples of these situations are:

- Roads where the curvature changes rapidly
- One or more of above information sources are missing
- One or more of above information sources are inaccurate

In these situations it would be beneficial to use other sources of information. With a radar sensor, stationary objects can be detected, examples of these are posts, fences and buildings. This information can be used to estimate where the road is likely *not* to be, rather than where it is. While the stationary objects provide "negative" information, the other listed information sources provide information about where the road is, which can be considered "positive" information.

Figure 1.2 is showing an example of a road where the curvature is changing rapidly and not much information about the change is available from the four listed sources.

Figure 1.2: Picture showing a road where the curvature is changing rapidly.

By combining many information sources and correlate them to each other it is possible to compensate for the weaknesses of every source. For example it may be possible to detect if one source is wrong if several others are reporting contradictory information. In cases where lane markings and barriers are providing faulty information, the use of stationary targets can help identify different failure modes in the system. Such as inaccurate lane marking observations. If a lane marking is heading straight into a stationary target, the observed lane is likely not accurate.

This thesis investigates possible methods to utilize this "negative" information in the process of creating a road model.

1.2 Aim

The aim is to derive and implement an algorithm for estimating the road geometry. The algorithm should utilize stationary objects, detected by a radar, in order to improve the estimation of the road geometry.

1.3 Objectives

In order to achieve the aim, the following objectives are set:

- Design an algorithm capable of modelling the road ahead using information from host vehicle motion, lane marking observations, barrier estimations and properties of vehicles in front.
- Extend the algorithm to be able to use information about single stationary objects, additional to those used for barrier estimation. This last information source should be easy to turn on and off in order to evaluate the benefit of using it or not.
- Evaluate the algorithm performance on logged data.

1.4 Limitations

The following limitations has been selected for this thesis

- The estimation of the geometry will only be evaluated with respect to longitudinal control applications, such as Adaptive Cruise Control. Not for lateral control applications, such as a highway pilot.

- The result is only evaluated at host vehicle speeds above 30 km/h.
- The development will be performed using MATLAB. The performance will be evaluated both by analysis in MATLAB and by driving the system on specific routes.
- Only data from a front sensing radar and camera will be used.
- The system will only be evaluated on European roads.
- Discrete-time models will be used.

1.5 Thesis outline

The thesis consists of five chapters. After the introduction, Chapter 2 presents the theory used as base to the developed algorithm. Chapter 3 presents the sensor information used and how the theories have been applied to the road geometry estimation problem. It describes how the mathematical models of the system is designed. In Chapter 4 the results of the performance evaluation is presented. The method used for evaluating the performance is also described. Finally, the discussion regarding problems with the implemented algorithm, future work and conclusions are presented in Chapter 5.

2

Theory

This chapter describes the underlying theories used in this thesis. It presents the methods used and also the most important equations of the implemented mathematical framework.

2.1 Properties of a road

New roads are constructed and designed according to certain rules and requirements. The organisation *Trafikverket*, is responsible for defining the requirements for Swedish roads [6]. These requirements can and should be used as starting point for designing a road estimation algorithm. The curvature is an example of such a property. The curvature is the inverse of the radius and is calculated as

$$C = \frac{1}{R}$$

where R is the radius of the road.

The maximal allowed curvature of a road is depending on the speed limit. In Table 2.1, a list of desired maximal curvatures on roads for different speed limits are presented.

Table 2.1: Examples of recommended maximal road curvatures for different speed limits [6].

Speed limit $[km/h]$	**Desired maximal curvature** $[1/m]$
60	$7.1 \cdot 10^{-3}$
80	$2.5 \cdot 10^{-3}$
100	$1.4 \cdot 10^{-3}$
110	$1.1 \cdot 10^{-3}$
120	$0.8 \cdot 10^{-3}$

There are also requirements for how quickly the curvature is allowed to change. If the curvature difference between the start and the end curvature of a road segment is larger than a limit value there has to be a transition curve. A transition curve is a clothoid connecting two road segments. In Table 2.2 the limit values for when a transition curve has to be used is presented.

Table 2.2: Limit values for curvature differences for different speed limits [6].

Speed limit $[km/h]$	**Maximal curvature change** $[1/m]$
60	$3.3 \cdot 10^{-3}$
80	$2.0 \cdot 10^{-3}$
100	$1.3 \cdot 10^{-3}$
110	$1.1 \cdot 10^{-3}$
120	$1.0 \cdot 10^{-3}$

In case a transition curve has to be used, there are rules to how the clothoid has to be designed. The clothoid has to be designed according to

$$L = A^2 \cdot C \tag{2.1}$$

where C is the curvature at the end of the road segment and L is the minimal length of the segment. The parameter A determines the minimal length of the clothoid and it should at least fulfill the values of Table 2.3. The change of the curvature should be done linearly along the transition curve, meaning that the curvature rate should be constant.

Table 2.3: Lower limit for the clothoid parameter [6].

Speed limit $[km/h]$	**Minimal clothoid parameter (A)** $[m]$
60	130
80	185
100	250
110	290
120	325

In Figure 2.1 a comparison of a curve with and without transition curve is made.

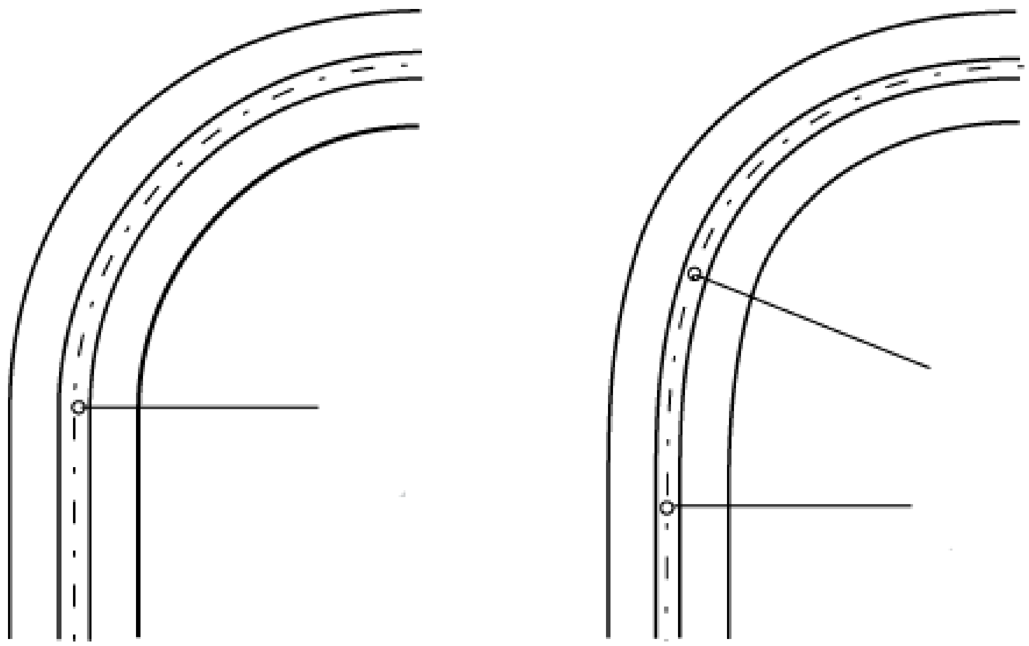


Figure 2.1: Comparison between curves without (left) and with (right) transition curve [6].

On public roads there are also connecting curves with opposing curvature directions, so called *S-curves*. These combination is also regulated by rules and requirements. Two opposing curves has to be connected with a straight line in between, which makes these combinations fall under the same requirements as the ones between a straight line and a single curve, Table 2.2.

All requirements listed in this section, such as maximum curvatures and curvature rates, can and should be considered when designing a road geometry estimation algorithm.

2.2 State representations

The state representation is used to mathematically describe the shape or geometry of the road in front of the vehicle equipped with the system, see Figure 1.1.

A road geometry can be represented in various ways. Examples of this are calculating the lateral position from a polynomial function of longitudinal distance, splines for interpolating between points or a clothoid model where curvature is a function of distance. It is common to use a coordinate system which is local, i.e it is fixed in some point of the car. In this thesis the local coordinate system will be fixed in the middle of the front of the car, the axes direction are according to the SAE-coordinate system [7], see Figure 2.2.

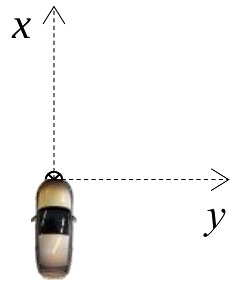


Figure 2.2: Definition of the coordinate system.

To use a polynomial, to represent lateral position from a function of longitudinal distance, is a way to represent relatively complex shapes with few arguments, for example a third degree polynomial is enough to be able to describe an s-curve shape. There are however drawbacks of using a polynomial of longitudinal distance, the polynomial cannot represent a change in the road shape far away, if for example the road is straight at close ranges and then turns. Additionally, it can only approximately represent a circular shape, which is a drawback since most roads are designed with a circular radius.

To solve the problem with representing a circular shape, a different model is used in [5]. This is called the clothoid model, which uses the following states; heading, curvature and curvature rate. This makes it possible to model a curve and the transition into or out of the curve. This model is also lacking in some scenarios of changing road curvature, if the curvature rate is not constant over the region modelled this model will have problems representing the shape [8].

The road geometry can also be represented with a sampled interval in the region of interest, i.e from zero to prediction range. This method is used in [3] and [4]. Which states that are represented in each point can be chosen in a number of ways. It can be position states, heading, curvature or curvature-rates in each point or a combination.

The angle of the road ϕ, is linked to positions according to

$$tan(\phi) = \frac{dy}{dx} \tag{2.2}$$

where x is the longitudinal position and y is the lateral position according to Figure 2.2. The curvature, C, is the distance derivative of the heading with regards to distance, i.e

$$C = \frac{d\phi}{ds} \tag{2.3}$$

where s is the distance along the road, i.e.

$$ds = \sqrt{dx^2 + dy^2} \tag{2.4}$$

Similarly, the curvature rate, κ, is the derivative of the curvature

$$\kappa = \frac{dC}{ds} \tag{2.5}$$

It is hence likely that the system will be overdetermined if a combination of the states are used in a sampled setting, for example the difference of two sampled curvatures is the curvature-rate multiplied by the distance between the two points. However, initial conditions are required on the lower level states, so if curvatures are sampled, initial conditions are required on position and angle to the road.

Below is a quick comparison between state representations. The road to be represented is consisting of four segments. The first segment is a 50 meter straight, the second is a 25 meter long transition curve, which is changing curvature linearly from 0 to 0.007 radians/meter, the third is a 50 meter curve with curvature 0.007 radians/meter and the final segment is a 25 meter straight. The shape of the curves is shown in Figure 2.3 and the position error as a function of distance is shown in Figure 2.4.

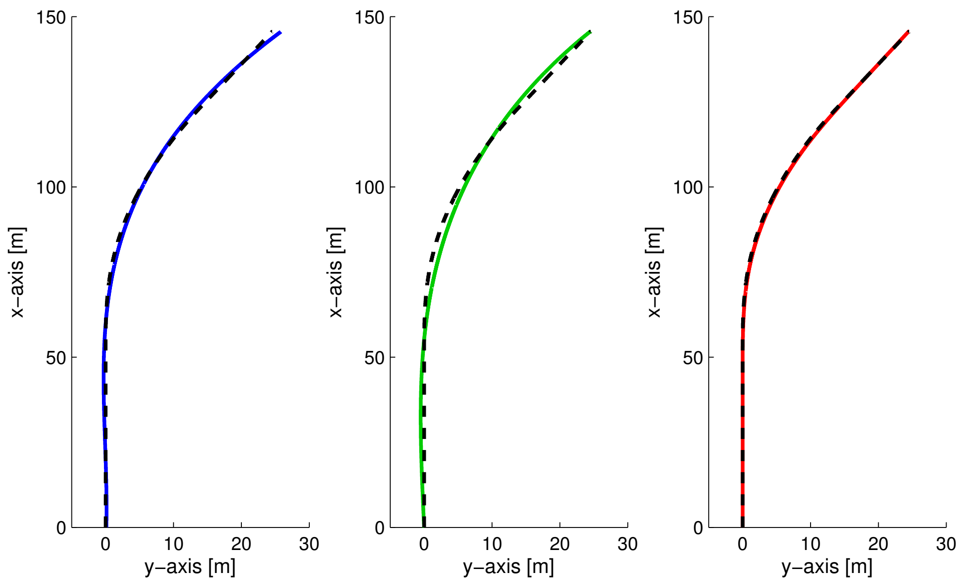


Figure 2.3: Comparison of state representations for an example scenario. The reference shape is the dashed black line, the blue line is a third degree polynomial representation, the green line is a clothoid representation and the red line is a sampled curvature representation

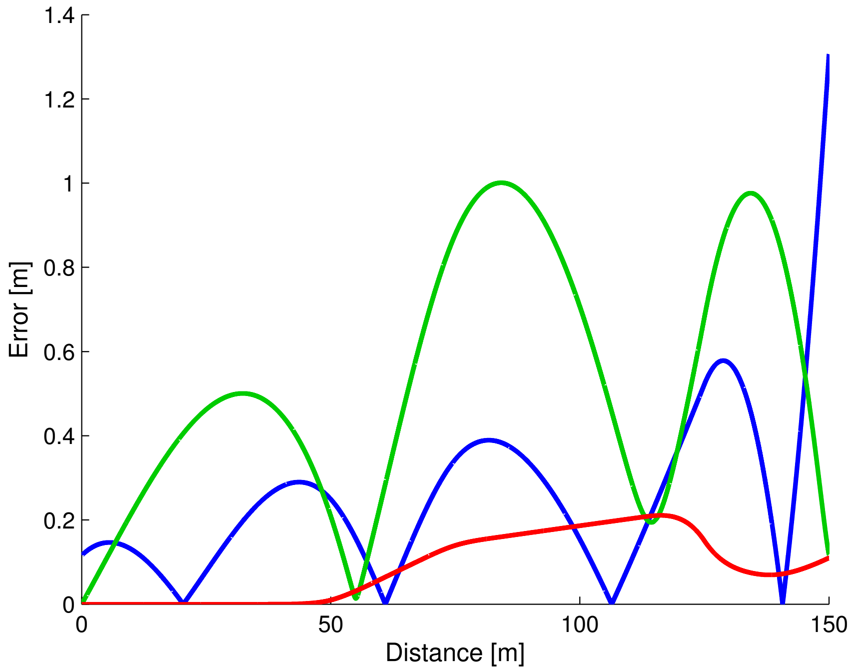


Figure 2.4: The absolute distance error for the state representations compared to the reference, the blue line is from a third degree polynomial representation, the green line is from a clothoid representation and the red curve is from a sampled curvature representation

All the states for the respective state representation was computed using a least squares method [16]. The sampled curvature model is using 20 samples in front of the vehicle. The mean error in this scenario was 0.25 meters for the polynomial representation, 0.53 meters for the clothoid model and 0.09 meters for the sampled curvature representation.

2.3 Statistical sensor fusion

This sections describes the basics of statistical sensor fusion and special variations of the standard methods used for modeling the road geometry. Bayesian statistics will be focused on.

2.3.1 Bayesian inference and filtering

The objective when performing inference of any kind is to be able to make decisions based on specific criteria. Bayesian inference use a probabilistic framework for performing these decisions, based on modeled uncertainties [9]. The applications of the framework is for example estimation or classification problems.

The Bayesian strategy base its methods on modeling the quantity of interest as a random variable with a probability distribution. The distribution is then updated according to acquired measurements and their uncertainties.

There are three major building blocks of the Bayesian strategy [9]. They are

- **Prior**
  The probability distribution of the unknown parameter before acquiring any observations. The notations commonly used for the prior is $p(x)$, where x is the unknown parameter.
- **Likelihood**
  This is the probability distribution for the observations, conditioned on the parameter x. This is usually noted $p(y|x)$, where y is the observation.
- **Posterior**
  The posterior is the probability distribution of the parameter given the observation. It contains the information obtained by combining the prior knowledge regarding the parameter and the acquired observations. It is calculated by using Bayes' rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \propto p(y|x)p(x) \tag{2.6}$$

  where the normalization factor is given as

$$p(y) = \int p(y|x)p(x)dx. \tag{2.7}$$

The above described strategy can be used to estimate the probability distribution of the state x, given the parameter y. For the cases where x is time dependent, this strategy can be modified and used recursively at every time step k. The *Prior* and the *Likelihood* will then be denoted $p(x_k)$ and $p(y_k|x_k)$ respectively.

The *Posterior* is then denoted $p(x_k|y_{1:k})$, where k is the current time-step of the process that increments as $1, 2, 3, etc.$ The sub-script of the parameter y, means the it is a vector containing all samples from time step 1 up to time step k. This recursive strategy of using the Bayesian framework can be used as a base for a filtering algorithm. In order to use the Bayesian framework for filtering, two distributions need to be modelled

- The predicted distribution of the states, also called the motion model. This describes how the unknown parameters are expected to change between time steps. This distributions is often denoted

$$p(x_{k+1}|x_k)$$

- The likelihood distribution, also called the measurement model. This describes where the observations are likely to be seen. This distributions is often denoted

$$p(y_k|x_k)$$

The objective when performing filtering is usually to compute $p(x_k|y_{1:k})$, which is the posterior distribution of the unknown parameter, given all observations up to the current time . By using the Bayesian framework above and Bayes' rule the joint posterior distribution for $x_{0:k}$ can be calculated as

$$p(x_{0:k}|y_{1:k}) = \frac{p(y_{1:k}|x_{0:k})p(x_{0:k})}{p(y_{1:k})} \propto p(x_0)\prod_{i=1}^{k} p(y_i|x_i)p(x_i|x_{i-1}) \tag{2.8}$$

In order to find the marginalized distribution for x_k, the variables $x_{0:k-1}$ need to be integrated out as

$$p(x_k|y_{1:k}) = \int p(x_{0:k}|y_{1:k})dx_{0:k-1} \tag{2.9}$$

The equations 2.8 and 2.9 describe the explicit use of full Bayes' rule and the complexity of the calculations increase with k. This property makes this method unfeasible in a real-time application [9]. The solution to this problem is to use a recursive method instead. In Figure 2.5, a block diagram showing how the recursive algorithm works.

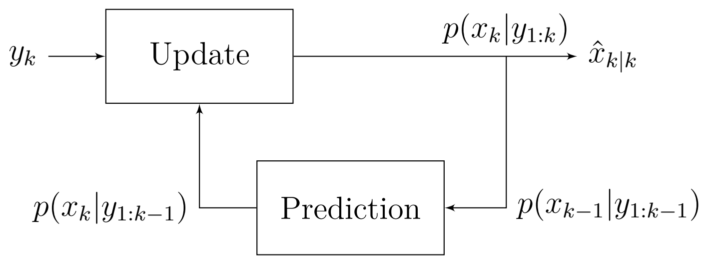


Figure 2.5: Block diagram of a recursive Bayesian filtering algorithm. y_k is the observation and $\hat{x}_{k|k}$ is the estimation of the unknown parameter at time step k.[10]

As can be seen in Figure 2.5 there are two calculations needed at every time step, the *prediction* and the *update*. The *prediction* needs to compute $p(x_k|y_{1:k-1})$ from $p(x_{k-1}|y_{1:k-1})$. It can be calculated using the conditional independencies between different time steps as follows

$$\begin{aligned} p(x_k|y_{1:k-1}) &= \int p(x_k, x_{k-1}|y_{1:k-1})dx_{k-1} \\ &= \int p(x_k|x_{k-1}, y_{1:k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \\ &= \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \end{aligned} \tag{2.10}$$

This equations is called the *Chapman-Kolmogorov equation* [9].

The *update* can be calculated as

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_k, y_{1:k-1}) \\ &= \frac{p(y_k|x_k, y_{1:k-1})p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \\ &= \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \end{aligned} \tag{2.11}$$

Equations 2.10 and 2.11 are general and can be applied to any filtering application.

2.3.2 The Kalman filter

There are few examples where it is feasible to use the filter equations mentioned earlier, as they normally do not have an analytical expression [11]. An important exception is when the prediction and update models are linear, the probability distributions are Gaussian and the noise is additive. In these cases, an analytical, closed form solution can be derived. This method was presented by Kalman in [12].

A linear and Gaussian model with additive noise can be expressed as

$$\begin{aligned} x_k &= A_{k-1}x_{k-1} + q_{k-1} \\ y_k &= H_k x_k + r_k \end{aligned} \tag{2.12}$$

where A_{k-1} is the *transition matrix* corresponding to the motion model and H_k is the matrix for the measurement model [13]. q_{k-1} and r_k are Gaussian random variables with covariance matrices Q_{k-1} and R_k respectively. These equations can be expressed using normal distributions as

$$\begin{aligned} x_k|x_{k-1} &\sim \mathcal{N}(A_{k-1}x_{k-1}, Q_{k-1}) \\ y_k|x_k &\sim \mathcal{N}(H_k x_k, R_k) \end{aligned} \tag{2.13}$$

By inserting the expressions in Equation 2.13 into Equation 2.10 and 2.11 an analytical expression for $p(x_k|y_{1:k-1})$ and $p(x_k|y_{1:k})$ can be derived. The commonly used and well known end result is presented below.

The *prediction* equations

$$\begin{aligned} \hat{x}_{k|k-1} &= A_{k-1}\hat{x}_{k-1|k-1} \\ P_{k|k-1} &= A_{k-1}P_{k-1|k-1}A_{k-1}^T + Q_{k-1} \end{aligned} \tag{2.14}$$

where P_k denote the covariance matrix for x at time step k.

The *update* equations

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k v_k \\ P_{k|k} &= P_{k_k-1} - K_k S_k K_k^T \end{aligned} \tag{2.15}$$

where the Kalman gain, K_k, the innovation, v_k, and the innovation covariance, S_k, are given by

$$\begin{aligned} K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ v_k &= y_k - H_k \hat{x}_{k|k-1} \\ S_k &= H_k P_{k|k-1} H_k^T + R_k \end{aligned} \tag{2.16}$$

The full derivation can be found in [9] and [13].

2.3.3 Extended Kalman filter

While the Kalman filter offers an optimal solution to the filtering problem in the case of linear and Gaussian models, many practical problems contain *non-linear* models, which makes the method not applicable. There are, however, methods to approximate the non-linear models as linear. One of which is called the *Extended Kalman filter* [14].

A non-linear model with additive noise can be written as

$$\begin{aligned} x_k &= f(x_{k-1}) + q_{k-1} \\ y_k &= h(x_k) + r_k \end{aligned} \tag{2.17}$$

where x_k is the state, y_k is the observation or measurement, q_{k-1} and r_k are Gaussian random variables with covariance matrices Q_{k-1} and R_k respectively, $f(\cdot)$ is the dynamic function for the system and $h(\cdot)$ is the measurement model.

The aim is to approximate the filtering densities used in the Kalman filter as Gaussian. This method uses Taylor series expansions to linearize the system model, $f(\cdot)$, and the measurement model, $h(\cdot)$, around the previous estimated state using the below relationships

$$\begin{aligned} f(x_{k-1}) + q_{k-1} &\approx f(\hat{x}_{k-1|k-1}) + f'(\hat{x}_{k-1|k-1})(x_{k-1} - \hat{x}_{k-1|k-1}) + q_{k-1} \\ h(x_k) + r_k &\approx h(\hat{x}_{k|k-1}) + h'(\hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1}) + r_k \end{aligned} \tag{2.18}$$

where $\hat{x}$ is the estimated state, $f'(\cdot)$ and $h'(\cdot)$ are the Jacobian matrices for the system model and the measurement model, respectively.

The resulting algorithm contains the equations below.

The *prediction* equations

$$\begin{aligned} \hat{x}_{k|k-1} &= f(\hat{x}_{k-1|k-1}) \\ P_{k|k-1} &= f'(\hat{x}_{k-1|k-1}) P_{k-1|k-1} f'(\hat{x}_{k-1|k-1})^T + Q_{k-1} \end{aligned} \tag{2.19}$$

The *update* equations

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k v_k \\ P_{k|k} &= P_{k_k-1} - K_k S_k K_k^T \end{aligned} \tag{2.20}$$

where K_k, v_k, and S_k are given by

$$\begin{aligned} K_k &= P_{k|k-1} h'(\hat{x}_{k|k-1})^T S_k^{-1} \\ v_k &= y_k - h(\hat{x}_{k|k-1}) \\ S_k &= h'(\hat{x}_{k|k-1}) P_{k|k-1} h'(\hat{x}_{k|k-1})^T + R_k \end{aligned} \tag{2.21}$$

A complete derivation can be found in [9].

3

Method

This chapter describes how the theories are applied to the road geometry estimation problem. It presents the available sensor information, its representation and how the information is used in the estimation process.

The scientific contribution of this thesis is a method or combining multiple sources of information in order to estimate the road geometry ahead of a host vehicle. The part which is the most interesting in the sense of contribution is the part where it is described how single stationary objects are used for the road geometry estimation. This is described under Section 3.2.3.4.

3.1 Sensor information

This section describes the different inputs used and their representations. The sensor used is a combined radar and camera sensor which can output its information via a User Datagram Protocol (UDP). In Figure 3.1, a picture of the sensor is shown.

3.1.1 Host vehicle motion

The sensor outputs the current speed of the host vehicle, $v_{host,k}$, and the current yaw rate, $\dot{\psi}_k$. These signals are assumed to have negligible uncertainties and are considered as control inputs to the estimation process. The host vehicles side slip angle is also reported by the sensor. It will be denoted $\alpha_{slip,k}$. In Section 3.2.2, a description of how the signals are used can be found.

3.1.2 Lane marking observations

The camera system is able to detect up to four different lane markings, the two closest to the host vehicle and the two next to them. If they are present and

Figure 3.1: Picture of the combined radar and camera sensor used as input source to the estimation process.

detected, they are reported as a third-order polynomial equation as,

$$y_k^i = a_{0,k}^i + a_{1,k}^i x_k + a_{2,k}^i x_k^2 + a_{3,k}^i x_k^3 \tag{3.1}$$

where $i = [1, ..., 4]$ for the different lane markings, y_k is the lateral position and x_k is the corresponding longitudinal position according to the coordinate system in Figure 2.2. The values $a_{0,k}$, $a_{1,k}$, $a_{2,k}$ and $a_{3,k}$ are the coefficients of the polynomial.

Each polynomial has an associated maximal valid range, denoted $x_{l,max,k}^i$. In Figure 3.2 an example of observed lane markings is shown.

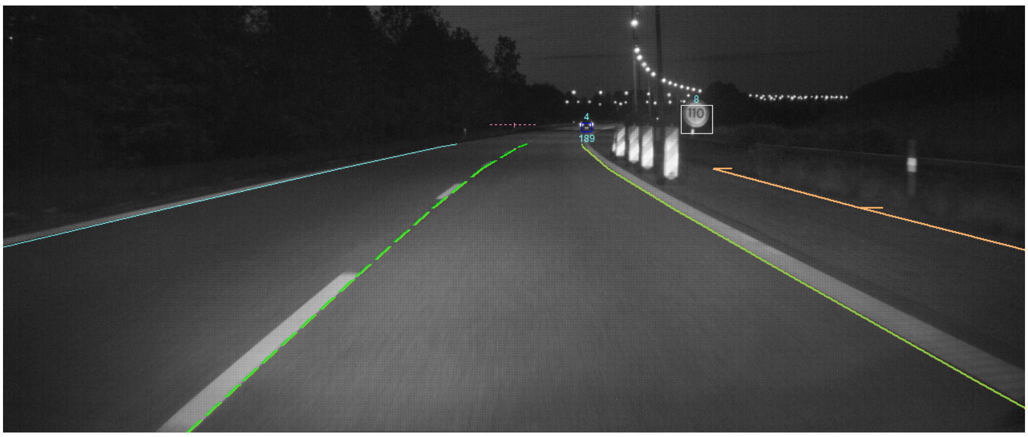


Figure 3.2: Picture showing an example of observed lane markings. It can be seen that the two closest, as well as an adjacent, lane marking are detected.

3.1.3 Barrier/guard-rail observations

The barrier/guard-rail observations are estimated by the sensor, using series of stationary radar detections along the road. The system has the possibility the detect one barrier on each side of the road. When present and detected, they are reported with a third-order polynomial in the same way as the lane markings as,

$$y_k^i = b_{0,k}^i + b_{1,k}^i x_k + b_{2,k}^i x_k^2 + b_{3,k}^i x_k^3 \tag{3.2}$$

where $i = [1, 2]$ for the different lane markings, y_k is the lateral position and x_k is the corresponding longitudinal position, according to the coordinate system in Figure 2.2. Here $b_{0,k}$-$b_{3,k}$ are the coefficients of the polynomial. The barrier-polynomial has, similarly to the lane markings, a maximal valid range, $x_{b,max,k}^i$. In Figure 3.3 an example of detected barriers is shown.

Figure 3.3: Picture showing an example of detected barriers. The green lines represent the reported barrier estimations.

3.1.4 Moving vehicle observations

The radar and the camera are providing information regarding detected objects and a fusion algorithm is estimating their motion. The motion states used from the objects are the longitudinal position, $x_{m,k}^i$, lateral position, $y_{m,k}^i$, heading, $\phi_{m,k}^i$ and velocity over ground, $v_{m,k}^i$, for the i^{th} detected object. All the object motion states are reported from the radar and camera fusion system with a corresponding estimate error standard deviation, $\sigma_{x,m}$, $\sigma_{y,m}$, $\sigma_{\phi,m}$ and $\sigma_{v,m}$.

A maximum of 64 objects can be detected by the fusion system and possibly be used for road geometry estimation.

3.1.5 Stationary observations

The stationary observations are reported by the radar. The properties of interest to the estimation process are their positions, the longitudinal position, $x_{s,k}^i$, and the lateral position, $y_{s,k}^i$. A maximum of 64 stationary observations can be reported.

3.2 System models

This section describes the details of all the models used for the estimation process. The way the states are represented and how they are stored is presented, as well as the process model used for time updates. The measurement models for different information sources are presented in their individual subsections. The method of detecting outliers is also described in this section.

3.2.1 State representation

The state representation implemented in the Kalman filter is a sampled curvature at fixed distances along the predicted road. An initial angle between the host vehicle and the road is also included as a state. The lane width and offset to the lane center is only estimated by the vision systems reports of the lane limits, they are therefore not included in the state vector. This is due to no other reliable way to estimate the lane limits has been derived.

The state vector is then

$$x_k = \begin{bmatrix} \phi_k \\ C_{0,k} \\ C_{1,k} \\ \vdots \\ C_{n,k} \end{bmatrix} \tag{3.3}$$

where ϕ_k is the angle between the host vehicle and the road and $C_{0,k} - C_{n,k}$ are sampled curvatures, all at time k. The curvature is starting from the host vehicle position, i.e in the origin of the coordinate system in Figure 2.2. The constant n is the number of sampled points in front of the host vehicle and corresponding to the number of subsections. The distance, δ, between samples is

$$\delta = \frac{PredictionHorizon}{n} \tag{3.4}$$

In Figure 3.4 a graphical representation of the states is shown.

There are dependencies of the positions of the estimated road, the position estimates are calculated according to

$$y = \int_0^{dist} \sin(\phi(s))ds \tag{3.5}$$

$$x = \int_0^{dist} \cos(\phi(s))ds \tag{3.6}$$

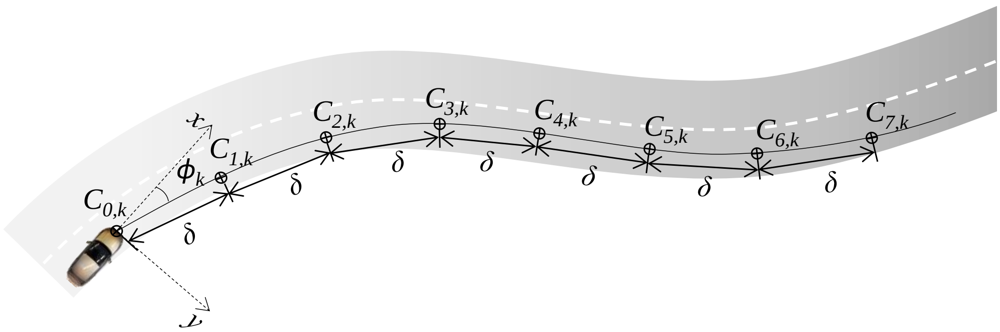


Figure 3.4: Illustration of the state representation in the local coordinate system.

where

$$\phi(s) = \phi_0 + \int_0^s C(x)dx \tag{3.7}$$

and $C(x)$ is piecewise linear according to the states in the state vector. The *dist* argument is chosen according to the interest of the application. The positions are numerically calculated where they are used in the road geometry estimation.

In the implementation used, $predictionHorizon = 200$ meters and $n = 40$.

3.2.2 Process model

The process model is accounting for the motion of the host vehicle and the evolution of the road. Since the states describe the road at fixed distances from the host vehicle and not fixed on the road, the actual points described is changed when the host vehicle is moving. The distance moved is simply computed by a constant velocity model.

$$ds = dt \cdot v_{host} \tag{3.8}$$

Where dt is the time since the last sample. It is then assumed that the curvature is changing linearly between the sampled points. This is described by

$$C_{i,k} = C_{i,k-1} + \frac{C_{i+1,k-1} - C_{i,k-1}}{\delta} \cdot ds = (1 - \frac{ds}{\delta})C_{i,k-1} + \frac{ds}{\delta} \cdot C_{i+1,k-1} \tag{3.9}$$

and the linear change is illustrated by figure 3.5.

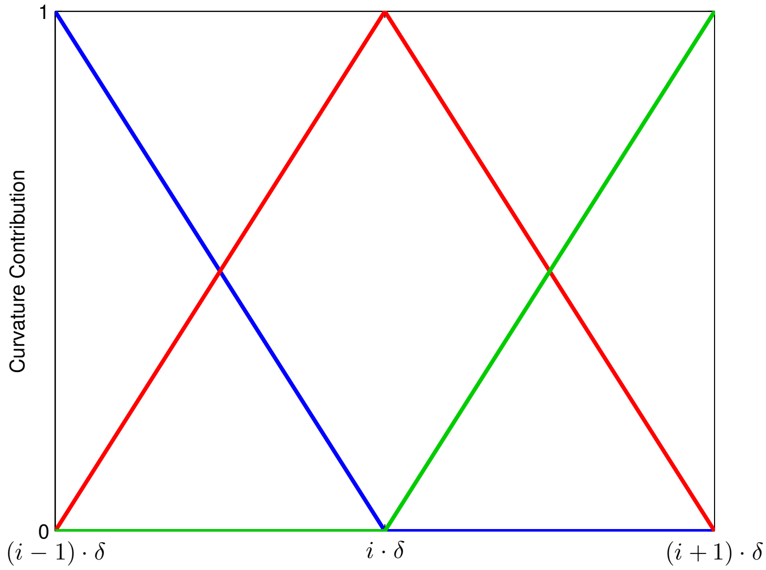


Figure 3.5: Curvature contribution as a function of distance from the host vehicle for the curvature states corresponding to distances $(i-1)\cdot\delta$ (Blue), $i\cdot\delta$ (red) and $(i+1)\cdot\delta$ (Green).

Assuming that $ds \leq \delta$ and for any index $i < n$, the last curvature is assumed to be constant. Rearranging the terms in equation 2.3, the angle is changing according to

$$\phi_k = \phi_{k-1} + ds \cdot C_{0,k-1} \tag{3.10}$$

If Δ is defined as

$$\Delta = \frac{ds}{\delta}$$

the transition matrix becomes

$$A_{k-1} = \begin{bmatrix} 1 & ds & 0 & 0 & \dots & 0 \\ 0 & (1-\Delta) & \Delta & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & (1-\Delta) & \Delta & 0 \\ 0 & \dots & 0 & 0 & (1-\Delta) & \Delta \\ 0 & \dots & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

It is also necessary to take the host turning into account. This is done using a constant turn model, where the host yaw rate is assumed constant between samples. The angle turned between samples is calculated by

$$\psi_k = d\psi_k \cdot dt \tag{3.12}$$

where ψ_k is the angle turned and $d\psi_k$ is the current host yaw rate. It is also assumed to have a small enough error to be used as an input signal to the system. The angle turned is directly taken from the initial angle state ϕ, resulting in an input

$$B\psi = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \psi \tag{3.13}$$

Additive uncorrelated process noise is also included in the model. This consists of two noise components, noise added to the angle state $q_k^{\phi} \sim N(0, \sigma_{\phi}^2)$ and noise added to each curvature state $q_k^C \sim N(0, \sigma_C^2)$. Written in vector form

$$q_k = \begin{bmatrix} q_k^{\phi} \\ q_k^C \\ \vdots \\ q_k^C \end{bmatrix} \tag{3.14}$$

This results in the complete update of the states

$$x_k = A_{k-1}x_{k-1} + B_{k-1}\psi_k + q_k \tag{3.15}$$

The process noise added, is calculated by

$$q_k^C = ((1 - noiseMin) \cdot noiseMax \cdot 0.5^{(\frac{v_{host}}{cutOffV})} + noiseMin \cdot noiseMax)^2 \cdot v_{host} \cdot dt \tag{3.16}$$

where $noiseMin = 5 \cdot 10^{-4}$, $noiseMax = 21 \cdot 10^{-4}$ and $cutOffV = 12$ which is the cut-off velocity. The noise component for the angle is calculated from

$$q_k^{\phi} = q_k^C \cdot ds^2 \tag{3.17}$$

How the noise is changing with the host velocity is illustrated in Figure 3.6.

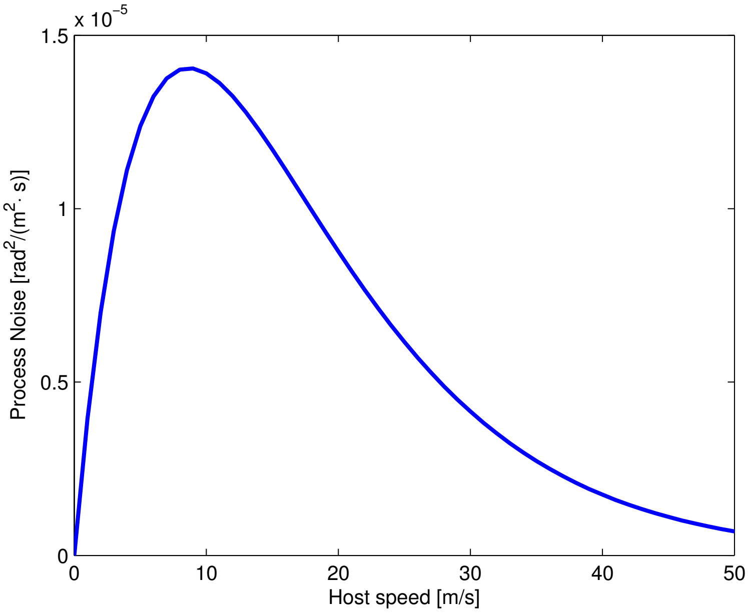


Figure 3.6: The magnitude of the curvature noise, q_k^C, added depending on the host velocity in $\frac{rad^2}{m^2 \cdot s}$, i.e before multiplying with dt.

3.2.3 Measurement models

In the framework for this filter, measurements of four different kinds are used, which come from the different information sources in section 3.1. The formats of the lane marker observations and the barrier/guard-rail observations are equal, so those are referred to as polynomials, hence only four kinds of measurements from five sources.

3.2.3.1 Host vehicle motion

Since the use case for the filter is longitudinal vehicle control applications, a driver is in charge of lateral control. It is then assumed that the driver is driving close to parallel to the road which is being estimated. The deviation from this assumption is modelled by measurement noise. So when the vehicle is going parallel to the road, the angle in which the host vehicle is moving, i.e the host slip angle α_{slip}, can be used as a measurement on the initial heading state ϕ_k. Similarly the curvature of the host vehicle motion is used as a measurement on the initial curvature $C_{0,k}$. This host curvature C_{host} is calculated by

$$C_{host} = d\psi_k / v_{host,k} \tag{3.18}$$

Using the notations in equation 2.12

$$y_k = \begin{bmatrix} \alpha_{slip,k} \\ C_{host,k} \end{bmatrix} \tag{3.19}$$

and

$$h_k = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \end{bmatrix} \tag{3.20}$$

The measurement noise covariance matrix of the host vehicle motion is modelled as

$$R_{host} = diagonal(\sigma_\alpha^2,\ \sigma_C^2) \tag{3.21}$$

where $diagonal(\cdot)$ is a square matrix with the input parameters on the diagonal, the measurement standard deviations are $\sigma_{k,\alpha} = 0.09$ radians and $\sigma_{k,C} = 0.003$ radians/m.

3.2.3.2 Lane and barrier polynomials

The lane and barrier detectors are both outputting the information in the form of a third degree polynomial. In order to utilize the information, it is transformed to a linear combination of the state vector. This means that the polynomial shape is estimated to angles and curvatures. Using what is known from equation 2.2, the angle in a point can be estimated from

$$\phi_{poly} = \arctan(\frac{dy}{dx}) \tag{3.22}$$

where $\frac{dy}{dx}$ can be calculated from equation 3.1 and 3.2 respectively. This results in

$$\begin{aligned} \frac{dy}{dx} &= a_1 + 2a_2 \cdot x + 3a_3 \cdot x^2 \Rightarrow \\ \phi_{poly} &= \arctan(a_1 + 2a_2 \cdot x + 3a_3 \cdot x^2) \end{aligned} \tag{3.23}$$

Furthermore to estimate curvature, the relation in equation 2.3 is used. Using the chain rule results in

$$C = \frac{d\phi}{ds} = \frac{d\phi}{dx} \cdot \frac{dx}{ds} \tag{3.24}$$

where

$$\frac{dx}{ds} = \sqrt{(}1 + \left(\frac{dy}{dx}\right)^2) \tag{3.25}$$

The resulting curvature estimate is then

$$C_{poly} = \frac{d\phi_{poly}}{ds} = \frac{2a_2 + 6a_3 \cdot x}{(1 + \frac{dy}{dx})^{\frac{3}{2}}} \tag{3.26}$$

So both curvature and heading estimates from the polynomial can be computed at a given longitudinal distance, x.

Each polynomial is reporting a valid range which is used to select the measurement points, the initial and final angles and curvatures are estimated and used as measurements. The initial heading and curvature are corresponding directly to the states ϕ_k and $C_{0,k}$. To compute an angle estimate from the states at a given distance, the initial angle is used together with the distance along the road to sum up the angle change from the curvatures. This is how the measurement matrix is being calculated for the angle measurement at the valid range. The calculation is using the assumption of constant curvature rate between samples

H = zeros(4,nStates);
H(1,1) = 1;
H(2,2) = 1;
H(3,1) = 1;
i = 1;
while *validRange* $> i \cdot \delta$ **do**
  H(3,i+1) = H(3,i+1) + $\delta/2$;
  H(3,i+2) = $\delta/2$;
  i = i + 1;
end
K = ($validRange$ - (i-1)$\cdot\delta)/\delta$;
H(3,i+1) = H(3,i+1) + (1 - $K/2)K \cdot \delta$;
H(3,i+2) = $(K/2)K \cdot \delta$;
H(4,i+1) = (1 - K);
H(4,i+2) = K;

Algorithm 1: Calculation of the elements in the measurement matrix, H, for a polynomial input

The indexation is (row,column) and starting at index 1. The initialization, H = zeros(rows,columns), means it is a matrix with only zeros and nStates is the number of states in the state vector. Note that K takes a value between 0 and 1.

The measurement noise of both the lanes and barriers are modelled with a white uncorrelated noise. The measurement noise covariance matrix for lanes, R_L, is modelled as

$$R_L = diagonal(\sigma^2_{L,\phi},\ \sigma^2_{L,C},\ 5 \cdot \sigma^2_{L,\phi},\ 5 \cdot \sigma^2_{L,C}) \tag{3.27}$$

where $\sigma_{L,\phi} = 0.1$ radians and $\sigma_{L,C} = 0.005$ radians/m. The measurement noise covariance matrix is similarly shaped for barriers

$$R_B = diagonal(\sigma^2_{B,\phi},\ \sigma^2_{B,C},\ 5 \cdot \sigma^2_{B,\phi},\ 5 \cdot \sigma^2_{B,C}) \tag{3.28}$$

but in this case $\sigma_{B,\phi} = 0.25$ radians and $\sigma_{B,C} = 0.060$ radians/m. There are two reasons for the larger values of the barrier measurement noise. First, the barrier estimator is inadequate in many situations and it is not outputting any reliability measure. Second, the barriers on the side of the road are not always completely parallel to the road of the host vehicle. The benefit of using barriers are that they often have a longer range and hence can give a better estimate further away.

3.2.3.3 Moving vehicle observations

Vehicles driving in front of the host vehicle can indicate the shape of the road further away than the range of vision lanes. From the vehicle information, the longitudinal position and heading of the vehicle are used. The angle is used as the measurement. The longitudinal distance is used to determine where the angle is valid. The angle estimate from the states is computed in a similar way as the ending angle measurement of the polynomials.

H = zeros(1,nStates);
H(1,1) = 1;
i = 1;
while *vehicle longitudinal position* $>$ *longDist(i·δ)* **do**
    H(1,i+1) = H(1,i+1) + $\delta/2$;
    H(1,i+2) = $\delta/2$;
    i = i + 1;
end
K = ($validRange$ - longDist((i-1)·δ))/(longDist(i·δ)) - longDist((i-1)·δ)));
H(1,i+1) = H(1,i+1) + (1 - $K/2)K \cdot \delta$;
H(1,i+2) = $(K/2)K \cdot \delta$;

Algorithm 2: Calculation of the elements in the measurement matrix, H, for a single preceding vehicle input

Where the longdist(·) function is calculating the longitudinal distance of the estimated road at the input distance along the road according to equation 3.6 and 3.7. The way the measurement matrix is computed is using an assumption that the left and right edge of the road has the same heading at a certain longitudinal distance, i.e the lateral distance does not matter.

The fusion system is outputting the standard deviation of the heading angle estimate of the car, $\sigma_{fus,heading}$. Because this is a filtered input to this system, the measurement variance is doubled.

$$\sigma^2_{heading} = 2 \cdot \sigma^2_{fus,heading}\ radians^2 \tag{3.29}$$

An additional noise is added because vehicles are not driving precisely parallel to the road, this veering is estimated to

$$\sigma^2_{veer} = 0.03\ radians^2 \tag{3.30}$$

The further away a car is, the more likely it is to be on a road which is not parallel to the road which is of interest. This applies in both longitudinal and lateral distance. The additional measurement noise added because of lateral distance to the road is

$$\sigma^2_{lat} = \frac{lateralDistance}{laneWidth} \cdot 0.07\ radians^2 \tag{3.31}$$

which means it is proportional to the number of lane widths away it is. The longitudinal distance noise is

$$\sigma^2_{long} = longitudinalDistance \cdot 0.003\ radians^2 \tag{3.32}$$

The resulting measurement noise is

$$R_{car} = \sigma^2_{heading} + \sigma^2_{veer} + \sigma^2_{lat} + \sigma^2_{long} \tag{3.33}$$

3.2.3.4 Stationary object observations

A stationary object observation can indicate that something is obstructing the road at the position where it is detected. This information can help the road geometry estimator to detect a curve ahead before there is any other information about the curve available. The information used from the stationary objects is their longitudinal and lateral position. The lateral position is used as a measurement and the longitudinal position is used to determine where the measurement is valid. The lateral position of the current states is calculated using equation 3.5.

The information obtained from the stationary objects are considered as "negative" information to the estimation process. They tell where the road is likely *not* to be. Therefore, a special model for the *likelihood* is needed, rather than the normally used model for positive information. The model used is expressed as

$$p(y|x) = \frac{1}{V}\,\mathcal{N}(0; 0, R) - \frac{\gamma}{V}\,\mathcal{N}(y; h(x), R) \tag{3.34}$$

where V is a normalizing factor, R is the measurements noise and γ is a factor between 0 and 1, determining how "negative" the information is. The resulting distribution will, for a one dimensional case, look approximately like the one in Figure 3.7.

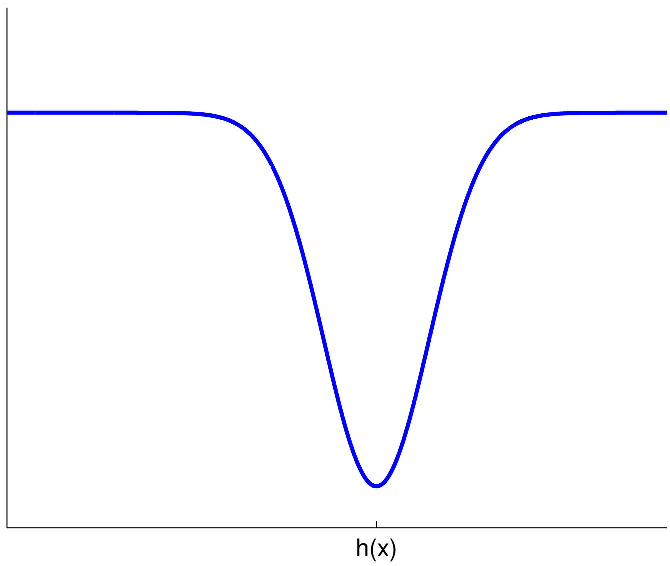


Figure 3.7: Example of how the likelihood distribution looks like for a stationary object measurement for a one dimensional case.

As can be seen, the likelihood for a measurement decreases around the current state. The area of the "negative" part is equal to the factor γ.

By using this model for the likelihood, the expression for the posterior becomes

$$p(x|y) \propto p(y|x)p(x) = \frac{1}{V}\,\mathcal{N}(0;0,R)p(x) - \frac{\gamma}{V}\,p(\bar{y}|x)p(x) \tag{3.35}$$

where $p(\bar{y}|x)$ is the negative part of the likelihood. Applying Bayes rule on the last term of equation 3.35 results in the following

$$p(x|y) \propto \frac{1}{V}\,\mathcal{N}(0;0,R)p(x) - \frac{\gamma}{V}\,p(\bar{y})p(x|\bar{y}) \tag{3.36}$$

where $p(x|\bar{y})$ can be considered as the posterior distribution of x when $\bar{y}$ is used as a measurement and $p(\bar{y})$ can, in the case of a linear measurement model, be expressed as

$$p(\bar{y}) = \mathcal{N}(\bar{y}; Hx, R + HPH^T) \tag{3.37}$$

However, the measurement model for the lateral position is non-linear. Therefore a version of an Extended Kalman Filter is used to process the measurement from the stationary objects. The measurement equations are then expressed as equations 2.20 and 2.21. It is necessary to derive an expression for $h'(x)$ in order to calculate both $p(\bar{y})$ and $p(x|\bar{y})$. When calculating the lateral position from the states, the equations 3.5 and 3.7 are used. However, to keep the complexity of the expression reasonable, it is assumed in equation 3.5 that the angle is piecewise constant over a distance of δ. The constant angle at the distance of $i \cdot \delta$ is calculated by

$$\begin{aligned}\phi(i \cdot \delta) = \phi_0 &+ \int_0^{\delta} C_0 + \frac{C_1 - C_0}{\delta} \cdot x \, dx \\ &+ \int_{\delta}^{2\cdot\delta} C_1 + \frac{C_2 - C_1}{\delta} \cdot x \, dx + \dots \\ &+ \int_{(i-1)\delta}^{i\cdot\delta} C_{i-1} + \frac{C_i - C_{i-1}}{\delta} \cdot x \, dx\end{aligned} \tag{3.38}$$

which is resulting in

$$\begin{aligned}\phi(i \cdot \delta) = \phi_0 &+ \frac{\delta}{2}(C_0 + C_1) \\ &+ \frac{\delta}{2}(C_1 + C_2) + \dots \\ &+ \frac{\delta}{2}(C_{i-1} + C_i)\end{aligned} \tag{3.39}$$

The constant angle estimate is then used to calculate the lateral position, the estimate at the distance $i \cdot \delta$ is calculated according to

$$\begin{aligned}h(x) = y &= \int_0^{\delta} \sin(\phi(\delta)) \, ds \\ &+ \int_{\delta}^{2\cdot\delta} \sin(\phi(2 \cdot \delta)) \, ds + \dots \\ &+ \int_{(i-1)\delta}^{i\cdot\delta} \sin(\phi(i \cdot \delta)) \, ds\end{aligned} \tag{3.40}$$

Using the assumption that the angles are piecewise constant results in

$$\begin{aligned}h(x) = y &= \delta \cdot \sin(\phi(\delta)) \\ &+ \delta \cdot \sin(\phi(2 \cdot \delta)) + \dots \\ &+ \delta \cdot \sin(\phi(i \cdot \delta))\end{aligned} \tag{3.41}$$

With the expression for $h(x)$ it is possible to calculate the Jacobian matrix

$$h'(x) = \begin{bmatrix} \frac{\partial h}{\partial \phi_0} & \frac{\partial h}{\partial C_0} & \frac{\partial h}{\partial C_1} & \dots & \frac{\partial h}{\partial C_n} \end{bmatrix} \tag{3.42}$$

When calculating the terms, it is necessary to know the measurement index of the stationary object. In the implementation used this index is calculated by

$$measIndex = \lceil (stationaryLongPos/\delta) \rceil \tag{3.43}$$

where $\lceil(\cdot)\rceil$ is rounding up to the next integer. For the initial angle, the partial derivative is calculated by

$$\frac{\partial h}{\partial \phi_0} = \sum_{k=1}^{measIndex} \delta \cdot \cos(\phi(k \cdot \delta)) \tag{3.44}$$

for the curvatures it is calculated according to

$$\frac{\partial h}{\partial C_i} = \frac{\delta^2}{2} \cdot \cos(\phi(i \cdot \delta)) + \sum_{k=i+1}^{measIndex} \delta^2 \cdot \cos(\phi(k \cdot \delta)) \tag{3.45}$$

for the curvature states where $2 \leq i \leq measIndex$. For the initial curvature

$$\frac{\partial h}{\partial C_0} = \sum_{k=1}^{measIndex} \frac{\delta^2}{2} \cdot \cos(\phi(k \cdot \delta)) \tag{3.46}$$

otherwise

$$\frac{\partial h}{\partial C_i} = 0. \tag{3.47}$$

for the curvatures where $i > measIndex$.

The resulting $h'(x)$ is then used to calculate $p(\bar{y})$ and $p(x|\bar{y})$. The distribution obtained by numerically calculating the complete posterior, using equation 3.36, is not Gaussian. For a one dimensional case it could look approximately like the distribution in Figure 3.8.

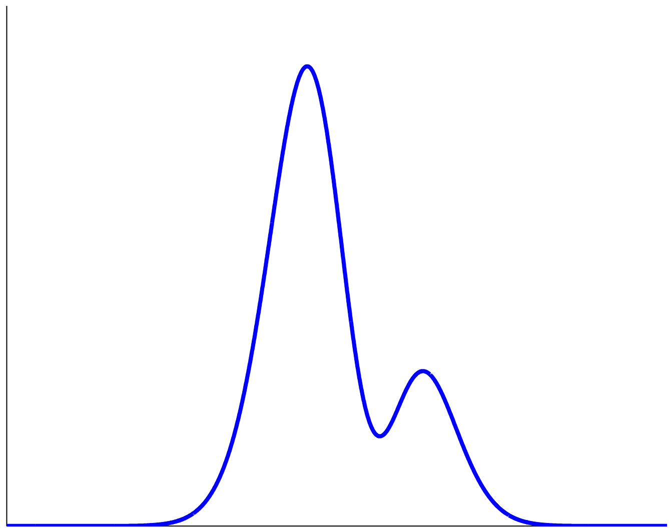


Figure 3.8: Example of how the posterior distribution can look like after a stationary object measurement update for a one dimensional case.

Since the state space used in the estimation process is not one dimensional, the calculation of the posterior becomes more complex. To calculate the full posterior numerically, all dimensions of the state space need to be "gridded" and $nGridPoints^{nStates}$ calculations would have been necessary. In the implemented algorithm that would result in more than 10^{70} calculations per measurement update, which is infeasible for a real-time system.

Instead, the posterior is calculated for one state at a time and the covariance between the states is disregarded. The non-Gaussian posterior is then approximated as Gaussian using the *Maximum Likelihood Estimation* (MLE)[17]. An example of an MLE is shown in Figure 3.9.

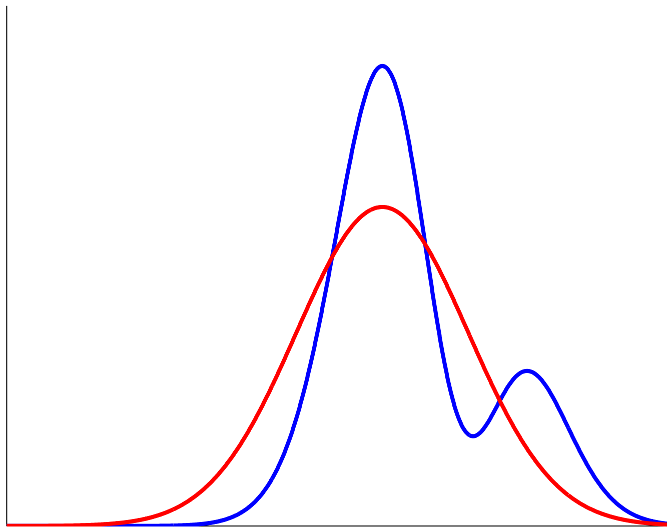


Figure 3.9: Example of how the MLE of the posterior distribution can look like. The blue line is a non-Gaussian distribution and the red line is its MLE distribution.

The correlation matrix for the states before the measurement update is stored and is then used to calculate the posterior covariance matrix. This way, the covariance of each state is updated with the stationary object measurements, but the correlation between the states stays the same. The number of necessary calculation is, with this method, reduced to approximately $nGridPoints \cdot nStates$. In the implemented algorithm, this results in 2091 calculations per measurement.

3.2.4 Outlier detection

This section describes the methods used for detecting outliers. The limits used to determine weather the measurement is an outlier or not is also presented.

3.2.4.1 Lane markings and barrier estimations

Since both lane markings and barriers are represented by a third degree polynomial, the same method of detecting outliers can be applied on both of them. The method chosen is to set an upper threshold on how much the lateral position at the valid range of the polynomials are allowed to differ from the estimated road model of the previous sample. The lateral error is compared to how long the polynomials are valid and the relation between the error and the valid range is used as a measurement. The mathematical expression is written as

$$Error\ value = \frac{|MeasEndLatPos - LatPosPrevPath|}{End\ distance} \cdot 100 \tag{3.48}$$

Since the error is calculated relative to the estimation from the previous sample, the method will contain a form of feedback. This may allow erroneous measurements to be used as long as they are consistent. However, when using multiple sources of information, the output from the estimation process is considered as the *best estimate* of the road geometry and is therefore used as a reference model for the outlier detection.

If the above calculated *Error value* is above a chosen threshold, the measurement is considered an outlier. In Figure 3.10 an example of the outlier detection for lane markings is shown. As can be seen, the threshold is set to 8 percent for lane markings and that value is exceed three times during the example log. The same threshold is set to 2 percent for the barrier estimations.

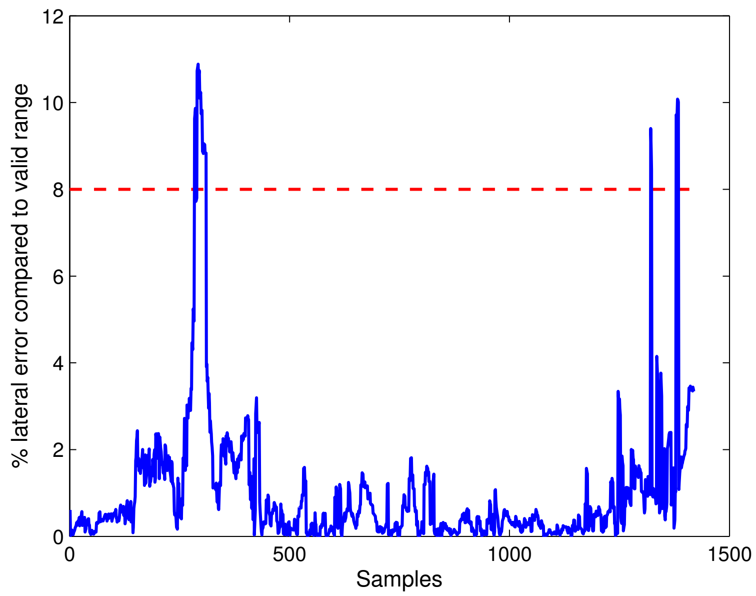


Figure 3.10: Example of lane marking outlier detection. The blue line is the lateral error as percent of the valid range of the lane markings compared to the estimated road position. The red line is limit used to determine whether the lane markings are outliers or not.

In Figure 3.11 an example of a case when the reported lane markings are considered to be outliers is shown. As can be seen, the reported lane markings (green lines) are not following the actual lane markings and are therefore incorrect when comparing them to the estimation of the fused road geometry.

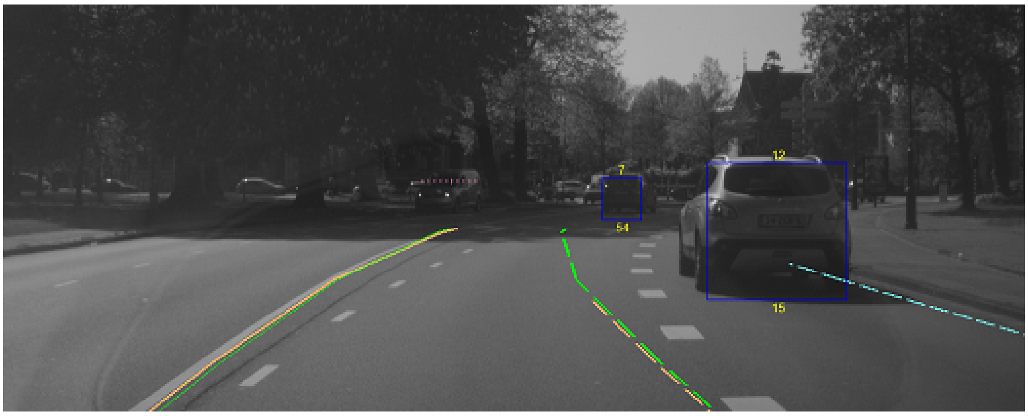


Figure 3.11: Example of a case when detected lane markings (green lines) are determined to be outliers.

3.2.4.2 Moving vehicle observations

To determine if the measurements of the preceding vehicles is outliers or not, the *Mahalanobis distance* is used. It can intuitively be interpreted as a measurement of how many standard deviations a point is from the mean of a probability distribution. It was presented in 1936 by P. C. Mahalanobis and is a commonly used measurement to reject outliers [15].

The *Mahalanobis distance* is calculated as

$$D_m(y) = \sqrt{(y - H\hat{x})^T (HPH^T + R)^{-1} (y - H\hat{x})} \tag{3.49}$$

where y is the point to which the *Mahalanobis distance* is calculated, $\hat{x}$ is the estimated state, P is the covariance matrix of the state, H is the measurement matrix and R is the covariance matrix of the measurement.

If the value calculated by equation 3.49 is above 1.5, the heading measurement of the preceding vehicle is considered as an outlier.

4

Results

The results of the road geometry estimation is presented in this chapter. The performance measures used to evaluate the algorithm is described, as well as the method of generating a reference of the road geometry.

4.1 Evaluation method

This section describes how the performance of the algorithm is measured. It describes how the reference is created and what measurement is used as a performance value.

4.1.1 Generation of ground truth

To be able to measure the performance of the estimated road geometry, a reference road geometry needs to be created. An accurate way of doing that would be to use a GPS and measure the position at each time step. However, in the used setup, there is no GPS available and the ground truth needs to be generated in an other way.

The chosen method is to use the measured speed and yaw rate of the host vehicle to estimate the movement a global coordinate system. This method is knows as dead-reckoning. In each time step the following calculations are performed.

The movement in the vehicles local coordinate system is

$$\begin{aligned} \Delta x_{host,k} &= v_k \cos(\frac{\dot{\psi}_k dt}{2})dt \\ \Delta y_{host,k} &= v_k \sin(\frac{\dot{\psi}_k dt}{2})dt \end{aligned} \tag{4.1}$$

where v_k is the velocity of the host vehicle and $\dot{\psi}$ is the current yaw rate. dt is the time elepsed from the previous sample.

The movement in the global coordinate system is

$$\psi_k = \psi_{k-1} + \dot{\psi}_k dt \tag{4.2}$$

$$\begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} \cos(\psi_k + \alpha_{slip,k}) & -\sin(\psi_k + \alpha_{slip,k}) \\ \sin(\psi_k + \alpha_{slip,k}) & \cos(\psi_k + \alpha_{slip,k}) \end{bmatrix} \begin{bmatrix} \Delta x_{host,k} \\ \Delta y_{host,k} \end{bmatrix} \tag{4.3}$$

where ψ_k is the current heading and $\alpha_{slip,k}$ is the host vehicles slip angle.

The delta distances is summed over time to get the ground truth road geometry. The dead-reckoning method will drift from the real world result over time. The error, however, is negligible if the geometry is not used than 200 meter at every time instance [3], which is a reasonable assumption for this application.

An example of the generated ground truth can be seen in Figure 4.1.

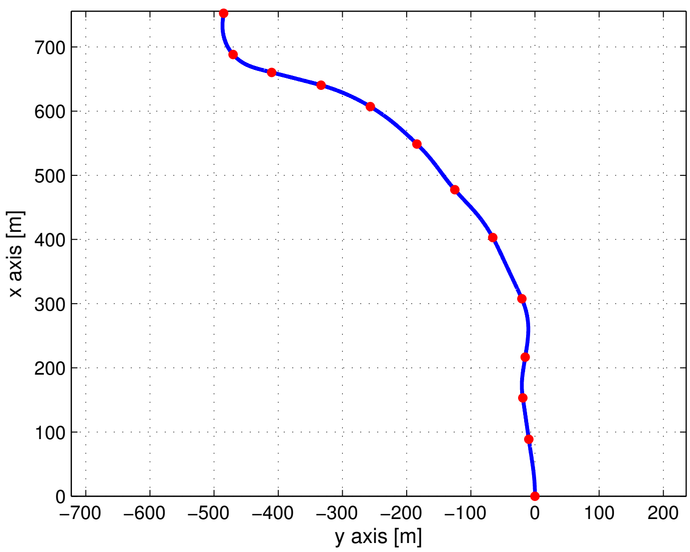


Figure 4.1: Examples of the generated ground truth in a global coordinate system. The red dots correspond to the position every 5 seconds.

4.1.2 Performance measure

Two measures are used to evaluate the performance. The first is the root-mean-square error (RMSE) between the estimated path and the ground truth path. The estimated and the ground truth positions are compared at headway times between 0 and 5 seconds, sampled every 0.1 seconds. This gives a measurement of the overall performance and indicates the behaviour over time.

In Figure 4.1 the ground truth for an example log file could be seen. The corresponding result of the position error evaluation can be seen in in Figure 4.2. The error shown is a 2 (blue) and 4 (red) seconds headway time. As mentioned earlier, the complete result is calculated between 0 and 5 seconds.

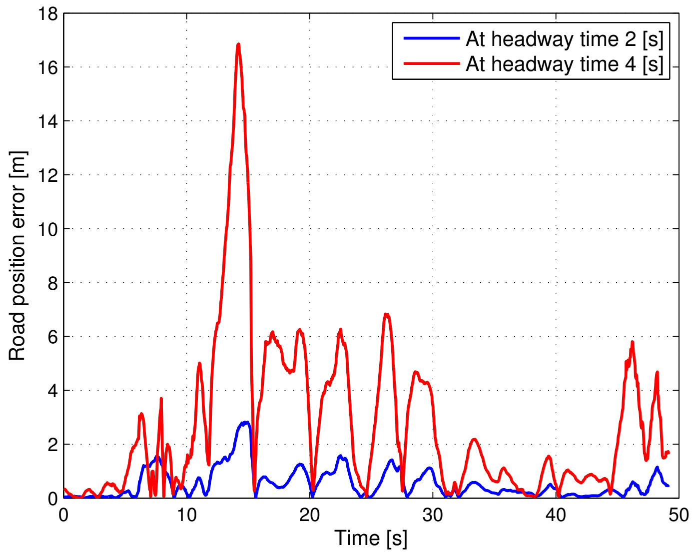


Figure 4.2: The position error at 2 (blue) and 4 (red) seconds headway time for an example log file.

The RMSE at different headway times is evaluated over time and is therefor presented as a result of a complete log file. The result of this analysis for the example log is shown in Figure 4.3.

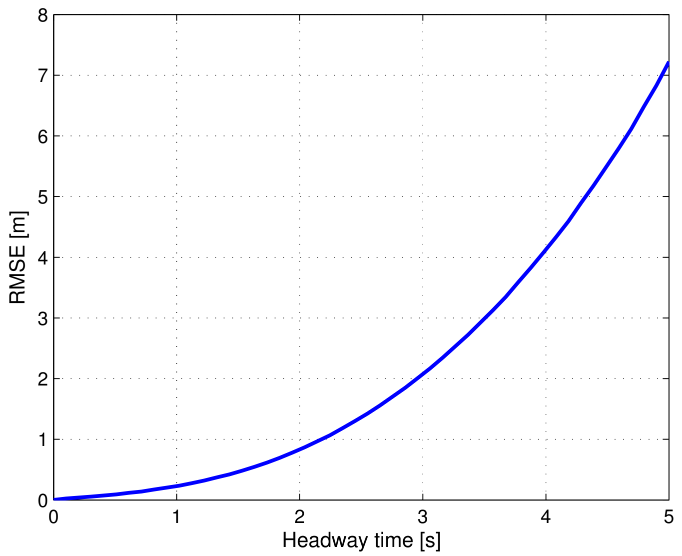


Figure 4.3: The RMSE of the road model estimate is the example log file.

The second measure is how much of the time the estimated path has an error less than the current lane width. The current lane width is estimated using the reported lane markers. Some of the functions utilizing the estimated road model need to be able to classify which lane objects are most probably in. This second measure then indicates how well the estimation would meet the needs of such a function.

The result of this second evaluation for the same log used in Figure 4.1 and 4.2 is shown in Figure 4.4. It can be seen that the estimated is never more than a lane width wrong for headway time under 2 seconds.

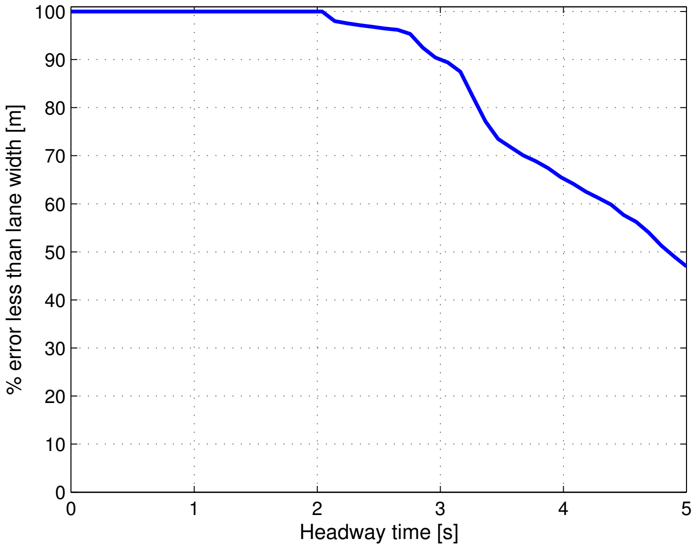


Figure 4.4: Percent of time the estimated road position is less than one lane width wrong at different headway times.

Results of the type presented in Figure 4.3 and 4.4 will be the measurement of how well the estimation algorithm performs in different traffic scenarios. These results can be studied in Section 4.2.

4.2 Algorithm performance evaluation

In this section the performance of the estimation algorithm is presented. Two different traffic situations are tested and the result for each of them is presented.

4.2.1 Curvy highway

The definition of a curvy highway is that the speed limit of the road is 100 km/h or higher and has multiple lanes. The "curviness" of the road is not specified with a number and the scenarios are selected based only on human judgement. Mountainous roads in the Alps are considered difficult in these type of applications and are therefore considered as suitable for this evaluation [3]. A picture showing an example of such a road can be seen in Figure 4.5.

Figure 4.5: Picture showing an example from a curvy highway.

The data set used for curvy highways is 420 seconds long and the average speed of the host vehicle is 27.3 m/s or 98 km/h.

As mentioned in Section 4.1.2, the first performance measurement used for evaluation is the RMSE at different headway times. In Figure 4.6 a comparison between three different methods of estimating the road geometry can be seen. The three methods are: using only lane markers, using lane markers, barriers and preceding vehicles and using lane markers, barriers, preceding vehicles and stationary object detections.

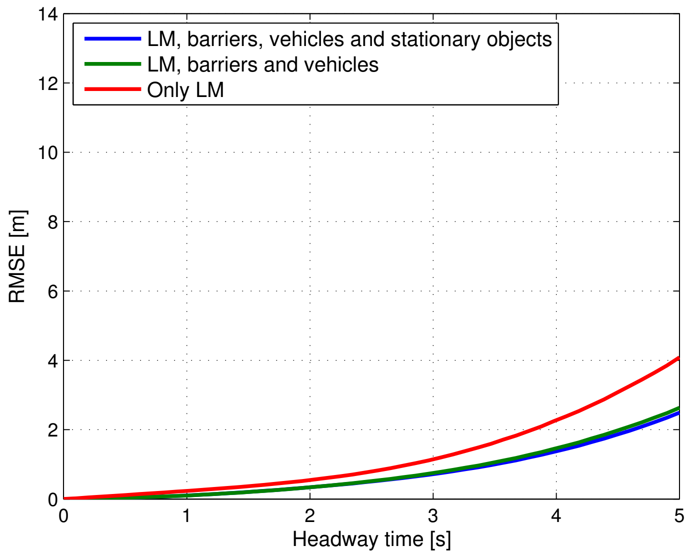


Figure 4.6: RMSE for the estimated road on the curvy highway data set.

As can be seen in Figure 4.6, the combination of different information sources improves the overall performance of the estimation compared to using only lane markers and results in a lower RMSE over the complete evaluation scope. It can also be seen that introducing the use of stationary objects improves the estimation slightly. However, the improvement is small enough to be caused by varying traffic scenarios in the selected logs.

The result of the second measurement of performance, the percent of the time the error estimation is less than a lane width, is presented in Figure 4.7. As can be expected, the estimation using multiple information sources is superior to using only lane markers also in this evaluation.

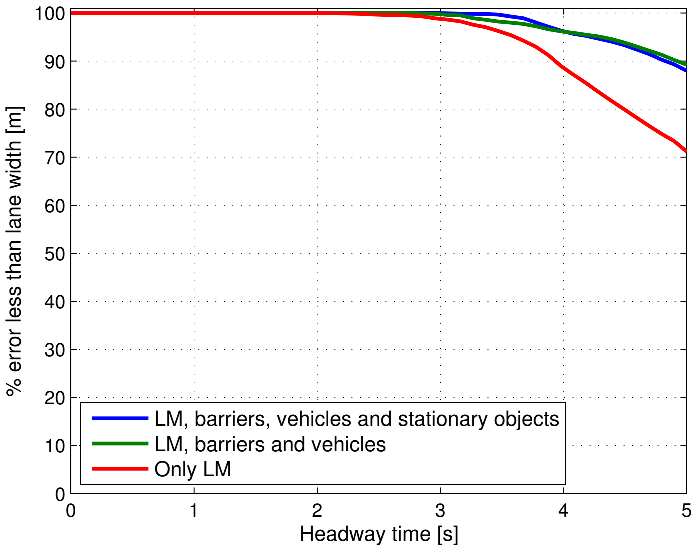


Figure 4.7: Percent of time the error of the estimated road position is less than one lane width on the curvy highway data set.

Even though the difference between using stationary objects and not using stationary objects is small also for this performance measure, it can be seen that the use of stationary targets makes the estimation error stay within one lane width for a longer headway time. With stationary objects, the estimation error is less than one lane width hundred percent of the time up to 3.5 seconds headway time, compared to just over 3 seconds for the method without stationary objects.

In the data set used, the lane markings are available 99.9 percent of the time.

4.2.2 Urban traffic

The definition of an urban road is that the speed limit of the road is between 50 and 90 km/h and has more than one lane. Typical examples of urban roads are arterial roads close to city centers. These roads cause difficulties for a road geometry estimator because of their rapid changes in curvature and relatively high speed limits.

In Figure 4.8 an urban traffic scenario is shown.

Figure 4.8: Picture showing an example from an urban road.

The data set used for evaluation on urban roads is 200 seconds long. The average speed of the host vehicle is 18 m/s or 65 km/h.

The RMSE for the road estimation is shown in Figure 4.9. As can be seen, the RMSE is significantly higher for the urban case than for curvy highway roads. This is to be expected and is due to the more rapid changes in curvatures.

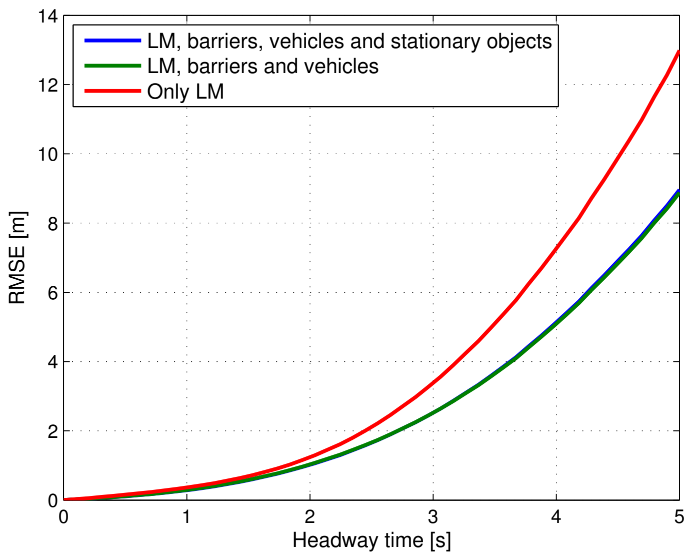


Figure 4.9: RMSE for the estimated road on the urban road data set.

Similar to the result on the curvy highways, the method which combine different information sources generates a better result than just using lane markers. The difference due to addition of stationary objects is, however, smaller in the urban traffic scenarios. This may be due to the amount of radar scatter picked up in a more complex traffic scenario, leading to more *false* detections of stationary objects that cause the road estimation to be erroneous. A *false* detection can, for example, be reflections from the ground or from overhead targets, such as traffic signs. These detections will be placed where the actual road is and when these are avoided by the road geometry estimator the result becomes incorrect.

In Figure 4.10 the percent of the time the error of the estimated road is less than one lane width is shown. Here, the affect of the *false* detections can be more clearly seen. The result of the method which uses the stationary object is worse than the method without stationary objects between 3.5 and 4.5 seconds of headway time.

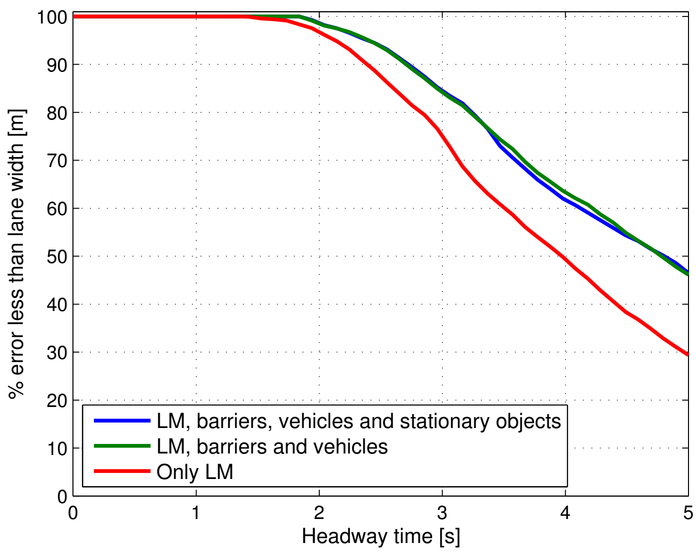


Figure 4.10: Percent of time the error of the estimated road position is less than one lane width on the urban road data set.

In the urban traffic data set, the lane markings are available 96 percent of the time.

5 Discussion

This chapter presents discussions regarding the implemented estimation methods and the results of the thesis. It also elaborates on the problems found during the implementation and evaluation of the algorithm. Suggestions to future work will also be stated.

5.1 Estimation methods

The state representation used can in most situations very well fit the shapes of the road. The fact that it can change the curvature multiple times over the prediction horizon is useful in some of the curvy urban scenarios. There is a trade off in how many sampled curvatures are used. The more they are, the better resolution is obtainable, but the more computationally heavy the filter is to run. The method of using sampled curvatures at fixed distances from the host vehicle, instead of them being at fixed points on the road, changes the transition matrix A. The split contribution to an estimated curvature between itself and the next curvature (non-zero elements on the diagonal and the superdiagonal), is causing a large covariance between states. This covariance cannot be tuned because it is inherent in the representation. The only benefit of keeping the curvatures at a fixed distance from the host vehicle is that the measurement models are slightly easier to compute.

The implemented method is heavily affected by the quality of the input from the sensors. Depending on the quality of every information source, the filter needs to calibrated accordingly to perform well. If all sensor information is erroneous, the estimation of the road geometry will also be wrong. A problem that has been noted during the development is that, if an information source is mostly reliable and the filters are calibrated according to the well behaving state of the sensor, the effect of bad, self-consistent, measurements is large and difficult to reject with for example outlier detection. A way to possibly avoid these behaviours would be to add quality indicators of the measurements from the sensors, and use these indicators to reject some of the inaccurate measurements.

Another aspect affecting the Bayesian filter setup is the fact that, all sensor measurements are filtered before they are received by the estimation algorithm. This makes the assumption of the Kalman filter, that all measurements are uncorrelated, to not hold. For this reason, the measurement covariances need to be adapted in order for the filter not to be too certain of its states. The amount of adaptation needed is very subjective and depends on the desired behaviour of the filter.

As the estimation algorithm uses the host vehicle motion, such as speed and yaw rate, the resulting road geometry estimate is only suitable for longitudinal vehicle control and not for lateral control. This is due to the fact that the lateral vehicle control would, by itself, have an impact on the host motion and would therefore create a self-feeding feedback, and may lead to instability and inadequate performance.

5.2 Results

The method chosen to generate the ground truth information is relatively easy to implement. It uses only sensor information already existing in the vehicle and needs no additional sensors, like a GPS. There are, however, a few drawbacks. The method contains systematic error due to the definition of the yaw rate of a vehicle. The yaw rate is the angular velocity with which the vehicle is turning around its vertical axis. When a road is banked, the angular velocity is not representing the vehicles motion in the global coordinate system used in the evaluation. This leads to an error in the generated ground truth model that will affect the result of the performance analysis.

Further, the result is always evaluated at headway times up to 5 seconds, even when there are no target vehicles present. In these cases it is not as important how accurate the estimation is. It is far more important that the estimation is correct when there are preceding vehicles present, as these are the objects in need of a lane assignment. The evaluation method could be adapted to include only samples where there are interesting targets present, and then the result will more accurately illustrate the performance perceived by a potential user.

The use of single stationary objects did not result in an enhanced performance of the road geometry estimation. It is common that stationary overhead targets are distorting the road estimate. Examples of stationary overhead targets are road signs above the road or bridges passing over the estimated road. Detections of these objects can then be placed in the middle of the road. A different problem encountered with the stationary objects is when the estimated road is on the wrong side of the object, the stationary object then pushes the estimate further out to the wrong side.

Another phenomenon noticed when evaluating the performance is that the result is highly dependent on the data set used for evaluation. An example is the difference between the result on curvy highways and the result on urban roads. This indicates that the result of the evaluation is difficult to compare if different data sets is used. The RMSE value of different algorithms evaluated on a different data sets is not directly comparable to each other. This is a weakens of the RMSE as a performance measure.

5.3 Future work

To improve the filter performance in the future there are some changes that can be made. The first suggestion would be to change where the sampled curvatures are fixed. So instead of being at fixed distances they are at fixed points on the road. Once a point is passed, a new point is placed at the prediction horizon distance. In this way there would be no correlation between states due to the transition matrix and the correlation can instead be tuned with the process noise model depending on the road type. It could also be beneficial to have a variable distance between sample depending on for example the host velocity. In low speeds, the prediction horizon does not need to be very long, but a high resolution is beneficial due to the curvature changing rapidly in some situations. The high velocity roads generally have much smaller curvature and curvature rate but a longer prediction horizon is needed. So by varying the distance between samples, a high resolution can be maintained in low speed while keeping a longer prediction horizon in high speed.

To improve the use of stationary objects, more information about them would be necessary. For example, in the current implementation there is no height information available from the stationary object. The height information could then be used to disregard those reflections that are likely to be from overhead objects or reflections from the ground. The problem with the estimated road ending up at the wrong side of the stationary objects could be be solved by having a better detector of which side of a stationary object the road is. For example, using the road width and where on the road the host vehicle is. An alternative solution is to not update with stationary targets unless being sure of which side of the road it is. A last suggestion is to use a multi-hypothesis filter to continuously evaluate both sides of the stationary objects and how likely each side is.

5.4 Conclusions

The use of single stationary objects is shown not to improve the road geometry estimation performance with the information currently available. Information about the objects elevation over ground could make it possible to disregard overhead targets and road reflections. An accurate barrier estimate, based on the stationary objects, would be more beneficial information for the road geometry estimation process.

The developed algorithm is shown to perform better than to use only lane markings as the road geometry estimation. The method is able to estimate the lateral position more accurately than the lane markings at headway times over 2 seconds. This makes it more useful for longitudinal vehicle control application, such as ACC, thanks to its higher potential for correct lane assignments of preceding targets.

Bibliography

[1] euroFOT (2012) Final results: Impacts on traffic safety, *European Large-Scale Field Operational Tests on In-Vehicle Systems*

[2] Rader, R. (2011) High-tech system on Volvos is preventing crashes, *IIHS* URL: http://www.iihs.org/iihs/news/desktopnews/high-tech-system-on-volvos-is-preventing-crashes

[3] Garcia-Fernandez, A. F.,Hammarstrand, L., Fatemi, M. and Svensson, L. (2014) Bayesian Road Estimation Using Onboard Sensors, *IEEE Transactions on Intelligent Transportation Systems*

[4] Fatemi, M., Hammarstrand, L., Svensson, L. and Garcia-Fernandez, A. F. (2014) Road Geometry Estimation Using a Precise Clothoid Road Model and Observations of Moving Vehicles, *2014 IEEE 17th International Conference on Intelligent Transportation Systems*

[5] Schiffmann, J.K. and Widmann, G.R. (1997) Model-based scene tracking using radar sensors for intelligent automotive vehicle systems, *IEEE Conference on Intelligent Transportation System. ITSC '97.*

[6] Trafikverket (2012) Krav för vägars och gators utformning, 2012:179, Borlänge

[7] SAE International, (1994), Sign Convention for Vehicle Crash Testing, J 1733

[8] Schwartz, D. (2003) Clothoid road geometry unsuitable for sensor fusion clothoid parameter sloshing, *Proc. IEEE Intell. Veh. Symp., pp. 484–488*

[9] Särkkä, S. (2013) Bayesian Filtering and Smoothing, Cambridge University Press, Cambridge

[10] Svensson, L. (2014), 'Optimal Filtering', SSY320, Chalmers University of Technology, Gothenburg, 31 August

[11] Svensson, L. (2014), 'The Kalman filter', SSY320, Chalmers University of Technology, Gothenburg, 1 September

[12] Kalman, R. E. (1960) A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering, 82:35-45*

[13] Gustavsson, F. (2012) Statistical Sensor Fusion, Studentlitteratur, Lund

[14] Jazwinski, A. H. (1970) Stochastic Processes and Filtering Theory, Academic Press

[15] Mahalanobis, P. C. (1936) On the generalised distance in statistics, *Proceedings of the National Institute of Sciences of India 2 (1): 49–55*

[16] Adams, R. A. (2013) Calculus: A Complete Course, Pearson Canada, Toronto

[17] Scholz, F. W. (2006) Maximum Likelihood Estimation. Encyclopedia of Statistical Sciences. 7.