Driving Manoeuvre Recognition and Prediction

Master of Science Thesis

ANDERS AGNVALL

Department of Signals and Systems
Division of Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2008
Report No. EX023/2008

Abstract

This thesis addressed the problem of recognising and predicting lateral driving manoeuvres in real-time based on measurements of vehicle movement and position in lane. An approach for modelling the driver-vehicle system was developed that uses a hierarchical driver model consisting of a tactical and an operational level. The tactical level of the model represented the manoeuvre intented by the driver whereas the operational level represented a set of control behaviours used to perform different manoeuvres. The model was applied to solve the problem of predicting driving manoeuvers based on early signs of lane change control behaviour.

A simulator study took place in order to collect data and evaluate the proposed prediction algorithm. It was shown that the algorithm on average was able to predict lane changes before any wheels crossed the line. The prediction performance was comparable to that of previous studies in the area. The algorithm needed no individual adaptation and was computationally efficient enough for real-time purposes.

This study took place at Volvo Technology Corporation at the department Humans, Systems and Structures.

Contents

Glossary and abbreviations

Table 1: Explanation of terms and abbreviations.

Term	Abbreviation	Explanation
Adaptive Cruise Control	ACC	An extension of a cruise control system that adapts to slower traffic and is able to keep a constant time gap to slower vehicles in front.
Forward Collision Warning	FCW	A driving support system able alert the driver in situations with imminent risk of a collision.
Automatic Emergency Brake	AEB	A driving support system able to identify siutations where a collision is unavoidable and if so automatically apply the brakes.
Lane Change Support	LCS	A driving support system alerting the driver if initiating a lane change when other vehicles are present in the blind spot or quickly approaching from behind.
Lane Departure Warning	LDW	A system intervening or alerting the driver if leaving the lane unintentionally.
Time to Line Crossing	TLC	Time until the first wheel crosses the lane marker given the current position and movement relative to the lane.
Electronic horizon	e-horizon	Database information used together with vehicle positioning in order to extract information about e.g. geography and infrastructure in a vicinity of the vehicle.

1 Introduction

Imagine that you are situated in a helicopter right above a car driving on a road. You have the ability to measure the speed of the vehicle accurately and you have a perfect view of the road and traffic in the surroundings. During such circumstances, would you be able to predict manoeuvres like changing lane, passing another car turning etc. based on the observed behaviour?

The problem would be different if you were located inside the vehicle together with the driver, being able to observe gestures and visual behaviour of the driver.

This report describes a method for classification and prediction of driving manoeuvers in a fixed base simulator. A system is presented that, by processing signals of vehicle data like lateral offset from the middle of the lane and speed recognizes and predicts lateral manoeuvres such as lane changes or takeovers. Longitudinal behaviour is discussed but not analyzed in this study.

The word *prediction* refers to recognizing in an early enough state of a manoeuvre so that the vehicle state in terms of lane position or change rate of lane position has not yet significally changed from that of driving without performing a lateral manoeuvre.

The development presented in this report is based on a driving simulator environment. However, it is kept in mind that the result should be transferrable to a real vehicle with real sensors in real traffic. Hence, no data is used that is not possible to measure in a real vehicle and the resulting algorithm must not be too computationally demanding to be executed on a vehicle ECU.

The report is organised as follows:

1. A background section explains the need for this investigation and the possible areas of application.
2. An approach for modelling a human–vehicle system is explained and discussed.
3. A simulator study aimed to test the approach is described.
4. Results are reported with comparison to another similar study (Pentlan and Liu [11]).
5. Discussion and recommended future work.
6. An Annex containing some additional information about the experiment.

2 Background

In the aim towards driving support systems, safer vehicles and safer traffic, a general desire is to be able to classify the state of the driver in different sences.

For example, in driving awareness research one would like to classify whether the driver is tired, awake, distracted or impaired for any other reason. For example

Pilutti and Ulsoy [12] made an experiment where the prediction error of a driver model based on an alert driver was compared to the actual output, a large difference between model predictions and measurements were taken as evidence that the driver was impaired. Piluttis approach is very similar to the one described in this report. Boer [2] used steering entropy as a measure for steering performance used as indicator of impairment. Other metrics such as mean square error of lane deviation [17], standard deviation of lane position [13], number of lane exits [17] and many others have been suggested as measures of vehicle handling performance.

Futhermore, in the development of driving support systems such as Adaptive Cruise Control (ACC), Forward Collision Warning (FCW) or Automatic Emergency Braking (AEB) it is essential to understand the driving situation in order to avoid erroneous system behaviour such as nuisance or missed interventions. An ACC systems requires the knowledge of which of the vehicles in front are in the intended path of the subject vehicle and could benefit from early knowledge about the intention of the driver.

FCW and AEB systems optimally requires knowledge about the intention of the driver since these systems should only activate if the upcoming situation is unintended. It is essential to have an as good as possible path prediction of the ego vehicle in order to understand which of several forward obstacles are in the path intended by the driver.

In Lane Change Support (LCS) systems, the driver is alerted if tending to change lanes while there are other vehicles present in the blind spot or quickly approaching from behind in the target lane. Today, such systems only know the intention of the driver to change lane by the turn indicator which is not always used or by the lane position or lateral velocity which are not necessarily evidence of an intention to move into the adjecent lane.

In Lane Departure Warning (LDW) systems it is of interest to know whether a lane departure is intentional or not, since nuisance alarms in such systems otherwise may become very frequent in certain driving environments.

The manoeuvre recognition system, if performing well enough, has potential of being applicable in path prediction and help providing earlier driver feedback in dangerous lane change situations. If it is able to separate intentional lane changes from unintentional ones, it would also be of use in lane departure warning systems.

3 Aim of this study

The goal with this study was to learn about driver modelling and driving manoeuvre prediction by trying to reproduce the results of previous research in the area such as the studies performed by Alex Pentland and Andrew Liu [11] and Nuria Oliver [10]. Furthermore, the aim was to develop an algorithm able to predict lateral manoeuvres with sufficiently good performance and a sufficiently low false rate to be able to improve the driving support functions mentioned in the previous chapter.

4 Modelling the Driver–vehicle system

The approach used for the prediction algorithm used in this study is to compare measurements of the vehicle state with predictions made by a set of driver models. The set of driver models, will each give input to a vehicle model in order obtain predictions of the vehicle state. The driver model that best predicts the next measurement at each time step will be used to conclude which manoeuvre is taking place at the moment.

Thus, as a first step, a model of the driver-vehicle system is needed. The driver model will general, with the oportunity to assign different behaviours to it, whereas the vehicle model will be static. The present chapter describes the overall driver-vehicle system. The following two sections describe the driver and vehicle models respectively.

4.1 System model

Figure 1 shows a block diagram over the system to be modelled. In the figure, F represents the driver and G represents the vehicle. The driver model is divided into two subsystems $F1$ and $F2$ where $F2$ represents a control behaviour and $F1$ represents a reference value fed into the rest of the system.

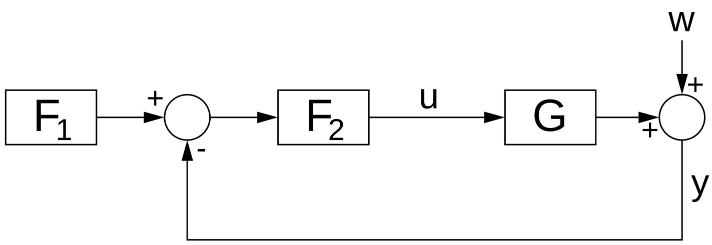

Figure 1: A model describing a driver–vehicle system. $F1$ represents the drivers intention and gives a reference value to $F2$ which represents the driver at an operational level. G is the vehicle dynamics.

A classical model such as the one illustrated in Figure 1 may be too simple to describe human behaviour. Pentland and Liu [11] suggested a hierarchical model where the driver is believed to use a set of different control behaviours depending on which the intended manoeuvre is. The set of control behaviours may contain different behaviours both with respect to $F1$ and $F2$. For example, in longitudinal control, two control behaviours may be car following and speed keeping. In the former case the driver is following a car at a suitable distance. In the latter case the driver has no other vehicle in front and is rather trying to keep a suitable speed considering the road characteristics. Even though both are control behaviours for longitudinal control they can be expected to have different characteristics not only in $F1$ but also in $F2$.

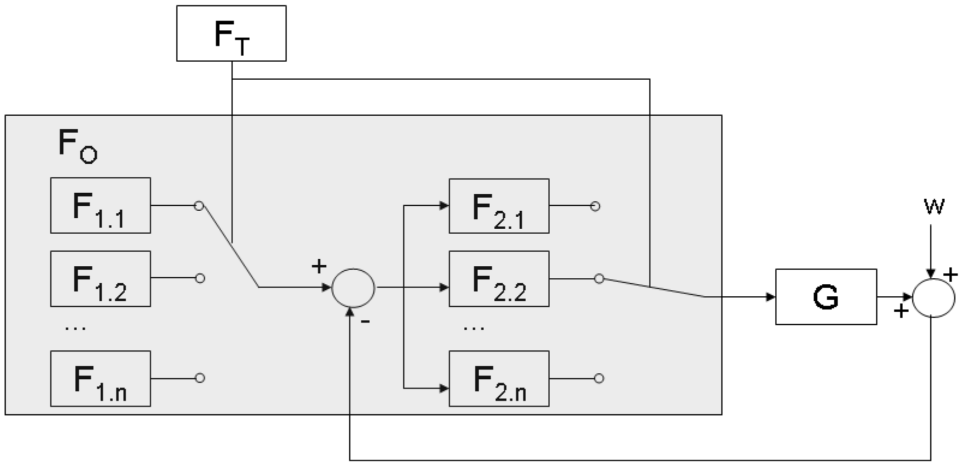

Figure 2: The driver is assumed to use different control behaviours $F1$ and $F2$, jointly denoted F_O, depending on what the intended manoeuvre F_T is.

Figure 2 illustrates the concept of the driving behaviour as being a composition of several control modes, each having different characteristics. The control behaviour is chosen by the topmost unit F_T 'T' meaning 'tactical', representing the intention of the driver. To fulfill a driving manoeuvre, the tactical level F_T of the driver chooses a series of control behaviours $F1,i$ and $F2,i$ (e.g. prepare, execute, conclude) that builds up the driving manoeuvre. The control strategies $F1,i$ and $F2,i$ are hereforth denoted F_O, 'o' meaning 'operational'.

This terminology derives from the Michon model [9] which is a three-level hierarchical model for driving behaviour. In the Michon model, the topmost level is the strategic level such as route planning. The second level is the tactical level including decisions such as speed-keeping, which lane to use, passing the vehicle in front or not etc. The third level of the Michon model is referred to as the operational level and includes the physical handling of the vehicle. In the model described in this section F_T can be seen as representing the tactical level and F_O the operational level of the Michon Model.

5 Vehicle dynamic model (G)

The Vehicle dynamics are meant to resemble those of the car simulator used in this study. The simulator is fixed base with a Volvo S80 as car mockup. The dynamics of the simulator were developed as a master thesis work and are closer described in [1]. However, in the present study, simple models extracted from log data are used as approximations which are good enough for the present purposes. Since only lateral behaviour is studied, only a steering dynamic model is created. No model representing the brakes or driveline is made.

5.1 Variables and coordinate system

The set of parameters used to describe the vehicle state and driver output is shown in Table 2. The choice of variables is influenced by the set of variables measurable in the car simulator.

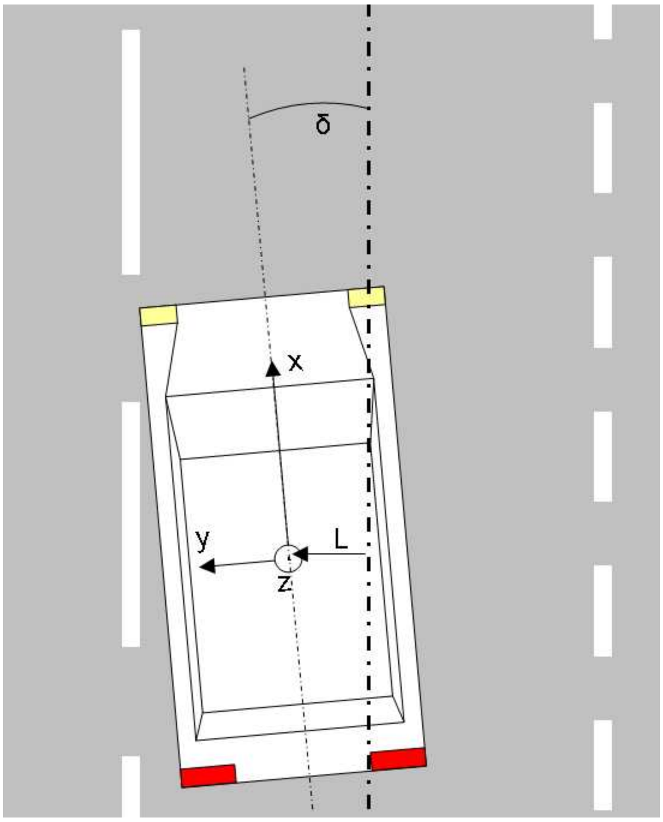

Figure 3: The coordinate system used in describing the vehicle is illustrated in the figure. Table 2 below explains the variables in more detail.

Table 2: System variables used to represent the vehicle state.

Parameter	Description
x, y, z	Vehicle based coordinate system [m]
v	Vehicle velocity in x direction (i.e. dx/dt) [m/s]
L	Lane position [m], distance from the centre of the current lane in perpendicular direction to the center of gravity of the vehicle. (Discontinous at lane change since the reference point moves to the centre of the new lane.)
δ	Angle of vehicle relative to the tangent of the road. [rad]
Θ	Steering wheel angle [rad]
ω	Yaw rate i.e. angular speed around the z axis [rad/s]
$\hat{}$	Denotes estimation of a parameter

5.2 Steering dynamics

The goal with the steering model is to create a model good enough for calculating the lane position L given the steering angle Θ. The exact steering dynamics are rather complicated since they depend on chassi and tyre details. The model presented here is a linear simplification that appears to be good enough for our purposes.

5.2.1 Estimation of yaw rate

Theoretically, the angular velocity ω (yaw rate) around a vertical axes, is dependent on the front wheel angle and the velocity. For our approximation we will assume that manoeuvres are moderate enough so that chassi dependent nonlinearities such as tire slip can be neglected.

We are thus searching a relation on the form

$$\omega = f(v, \Theta) \tag{1}$$

If one would drive with constant steering wheel angle, a lap would be completed in twice the time at twice the speed, so the angular velocity is clearly linearly dependent on velocity. Assuming that also the relation between wheel angle and steering wheel angle is linear we can guess that the relation we are looking for can be written

$$\omega = vg(\Theta) \tag{2}$$

where g is the function relating yaw rate to steering wheel angle. To reveal this function we plot yaw rate divided by speed against steering wheel angle from a simulator drive, see Figure 4. Shown in the plot there is also a least square linear curve fit. The figure indicates that the the function $g(\Theta)$ is linear but also that some other factor is present since the sample dots are a bit spread a little bit around the line. The spread data points may either represent noise or a higher order dependence. Since there is little noise in the simulator environment, the latter seems more reasonable.

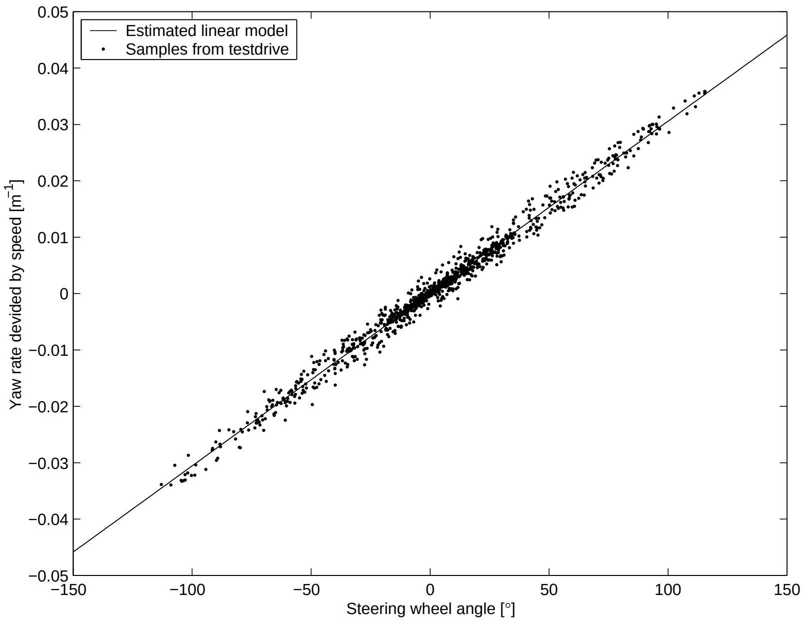

Figure 4: Yaw rate divided by speed plotted against steering wheel angle (dots) together with a linear curve fit (solid line). The relation between these variables seems to be linear although some other factor is also involved. The dotted values are measurements from a simulator test drive.

From Figure 4 we can conclude that the function

$$\hat{\omega} = Gv\Theta + \mu \tag{3}$$

where G is a constant and μ is the residual, is a reasonable first approximation to ω.

To investigate whether a second order depedence of steering wheel angle is involved the residual $\mu = \hat{\omega} - Gv\Theta$ is plotted against the time derivative of steering wheel angle in Figure 5.

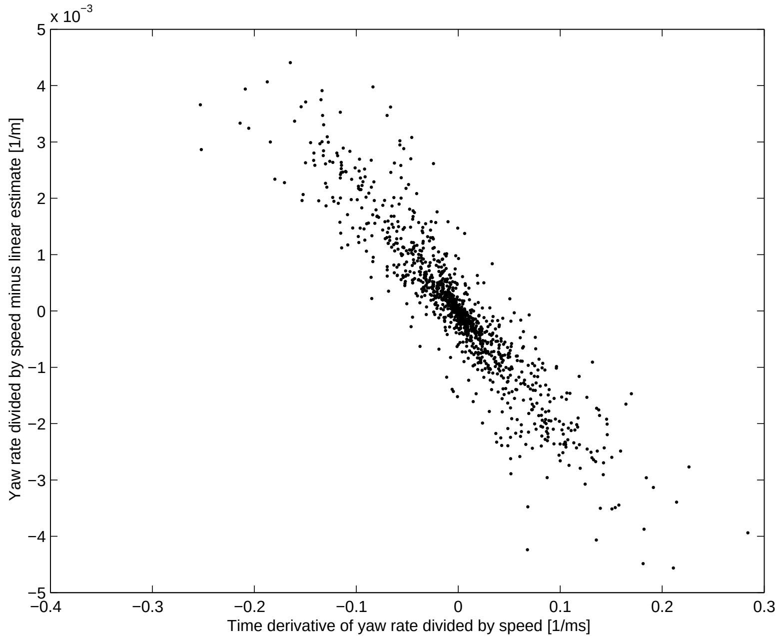

Figure 5: Difference between calculated and measured yaw rate as a function of the time derivative of the steering wheel angle.

Figure 5 indicates that the samples are somewhat gathered around a straight line. Assuming a linear relationship, the following expression applies:

$$\hat{\omega} - \omega \approx H\dot{\Theta} \tag{4}$$

where H is a proportinality constant that can easily be estimated via a least square approximation.

With $\mu = H\dot{\Theta}$ we now have a second order approximation of yaw rate according to

$$\hat{\omega} = Gv\Theta - H\dot{\Theta}, \tag{5}$$

For evaluation $v^{-1}\hat{\omega}(v, \Theta)$ is plotted against Θ in Figure 6.

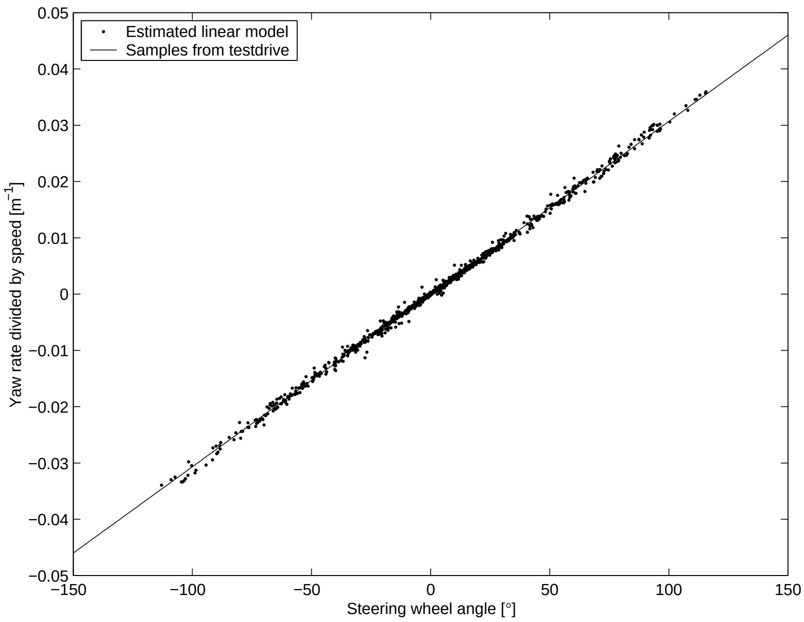

Figure 6: Yaw rate divided by speed calculated according to equation 5 against steering wheel angle together with measurements from a simulator drive.

As can be seen in the figure the measurements are closely gathered around the line obtained with our model.

5.2.2 Calculating lane position

We now have a model that describes what happens when we turn the steering wheel of the car. We now have to relate this to the coordinate system of the road. More specifically we want to be able to calculate the lane position L as a function of ω and thereby as a function of Θ. The origin of L is at the center of the lane that the vehicle is driving in. In this model we will assume that the road is straight. This means that a steering angle of 0 yields a constant value of the angle δ between the vehicle and the road. The road is not straight in the simulator world, but as will be explained later on in section 7, it is indeed sufficient to assume a straight road.

In order to calculate the lateral position L from steering wheel angle, we first need to calculate the heading angle δ. Assuming that the road is straight ω is the time derivative of δ. Thus, delta can be computed as

$$\delta = \delta(0) + \int_0^t \omega(t)\,dt \tag{6}$$

or in the discrete case

$$\delta_k = \delta_0 + \sum_{t=0}^{k} \tilde{\omega}_t\,\Delta t \tag{7}$$

Suppose now that we are travelling at angle δ compared to the road. Then the change in lane position during one timestep is

$$\Delta L = v\Delta t \sin\delta \tag{8}$$

Lane position at time $t = t_k$ follows as the sum of the initial lane position and the contributions at each time step up to time t_k.

$$L_k = L_0 + \sum_{n=0}^{k} \Delta L_{tn} \tag{9}$$

We have now derived an expression relating lane position to yaw rate. This expression together with Equation 5 constitutes a transfer function from steering wheel angle to lane position, thus providing the representation of the vehicle G in Figure 2.

5.2.3 Evaluation of steering dynamics

The figure below shows a comparison of measured lane position and lane position computed by the model G with measured steering wheel angle and speed as input.

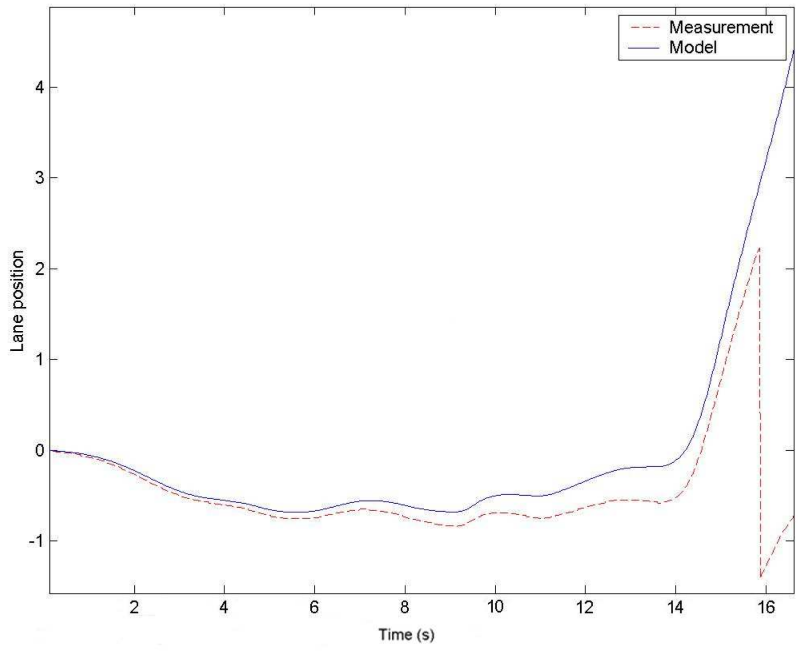

Figure 7: Comparison between model and the measurement. The model seem to be correct apart from a drifting error that grows along with time. The drift is probably due to road curvature not taken into account in the model.

The model follows the measurements reasonably well in the beginning. Later it drifts away due to an increasing error in the estimation of δ. This is due to the fact that there is some curvature on the road that we have disregarded from at the moment. However, the model will only be used to estimate the lane position small ΔT ahead, so the system will not be sensitive to such drift. In addition, as will be explained in the next chapter, the vehicle model will take the steering wheel of the driver simulation as input, not that of the real driver as in the comparison above. The simulated driver will be modelled as driving on a road with no curvature. Thus, road curvature can be disregarded from in this model.

At the right part of the figure there is a discontinuity in the measurements. This kind of discontinuity is a result of a lane change, since the origin of lane position L jumps from the previous lane into the new lane.

6 Driver dynamic model (F)

In this section, a model F of the driver is introduced. The model will be used to produce a steering wheel angle as input to the vehicle model given input variables such as lane position and vehicle speed. The goal with the driver modelling is to be able to resemble driver behaviour well enough to be able to identify different behaviour.

6.1 Driver modelling

The issue of modelling human vehicle lateral control has been widely studied before [12], [5], [7], [6], [11], [10]. There are several challenges with modelling a persons actions as a control system. A traditional control system is deterministic given the input signal and has certain behaviours that can be mathematically described. A person's actions, on the other hand, are subject to complex judgement based on many inputs. In addition, there is a great deal of individuality in a control task such as driving. Given exactly the same task to several drivers, they will solve the task in different ways.

One problem for example discussed in [12] is that a classic controller is an optimiser whereas a real driver rather acts like a *satisficer* e.g. a control system wants to stay exactly at a reference value given by some external unit. A real driver on the other hand tries to stay within a satisfying region of road. The region of satisfaction varies depending on surrounding traffic, environment etc. A human driver typically chooses a trajectory based on risk assessment, while taking also comfort and efficiency into account. As an example of risk consideration, if there is a large field of grass to the right of the road and there is busy traffic in the lane to the left the reference will be biased towards the field of grass. The reference value of a drivers lateral control can be seen as a corridor with time dependent width and center point (see [4]).

Human drivers tend to steer with soft movements in order to drive comfortably.

Typically, hard braking, hasty steering wheel movements or high values of yaw rate are avoided. Since yaw rate is proportional to the steering wheel angle multiplied with vehicle velocity, one could expect that the amplitude of steering is strongly related to speed.

May authors e.g. [16] have suggested predictive models for lateral control i.e. assuming that the driver is able to predict the vehicle state an amount of time ahead and base corrective actions on the predicted state of the vehicle rather than the current state.

Moreover, as part of the satisficer behaviour, a driver does not continuously evaluate the need for a corrective action. As seen in Figure 8 the steering wheel angle is typically piecewise constant with steering wheel reversals in between.

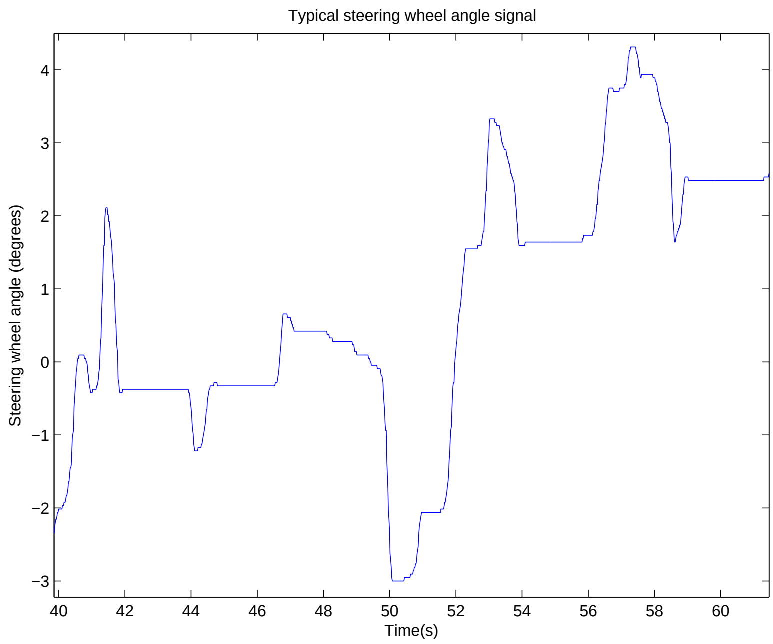

Figure 8: The image shows a part of the steering wheel angle log from one of the test subjects in the study.

In this study we need an as simple as possible dynamic model for lateral control which captures as much as possible of the human behaviour. The model described in the following section has proven to be a good candidate for such a model.

6.1.1 Operational driver model (F_O)

This section describes the driver model used at the operational level, i.e. the multiple control behaviours F_{Oi} introduced in chapter 4.1. When trying to find a suitable dynamic model for steering behaviour we are looking for a function relating the steering wheel angle to the vehicle state. Assuming again that the road is straight the most important inputs can be assumed to be the lane position, the angle to the road tangent and the vehicle velocity.

$$\Theta = f_{steer}(L, \delta, v) \tag{10}$$

Assuming that, according to the predictive behaviour discussed above, the driver uses only a predicted state L_{pred} of the vehicle lane position rather than the current state, the following applies:

$$\Theta = f_{steer}(L_{pred}) \tag{11}$$

where L_{pred} is the predicted lane position at time $t + \Delta T_{pred}$ based on the assumption that the angle δ remains constant. As illustrated in Figure 9, this is the lateral component of the motion vector yielding Equation 12, where the expression

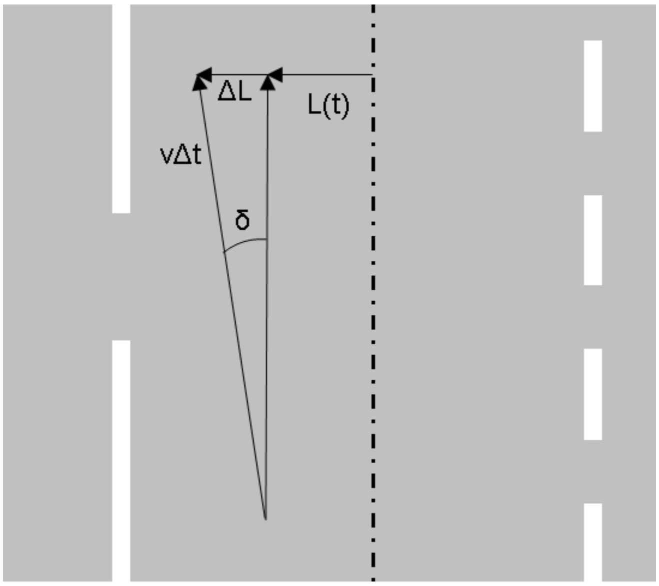

Figure 9: The driver model uses a prediction of lane position at ΔT ahead.

$$L_{pred}(t) = L(t) + vsin\delta\Delta T_{pred} \tag{12}$$

gives the predicted lane position ΔT_{pred} seconds ahead. The simplest possible controller would be one that steers to minimize the gap between the reference value L_{ref} and the predicted lane position.

$$\Theta(t) = -G_{steer}(L_{pred}(t) - L_{ref}) \tag{13}$$

G_{steer} in the equation is the controller gain.

Referring to the discussion in chapter 6.1, it is desirable that the model act as a satisficer. However, the model described by Equation 13 is, strictly speaking, an optimiser. Instead of letting the model control the steering wheel angle on the error $(L_{pred}(t) - L_{ref})$ directly, a deadband or another kind of cost function could be used in order to give the model a weaker response to small lateral deviations. However, the model of Eqution 13 seemed to perform well for the present scope. In fact, the model still captures the satisficer behaviour to some extent as a result of its predictive behaviour. It may be *satisfied* with a small lateral offset as long as it is heading towards the road center.

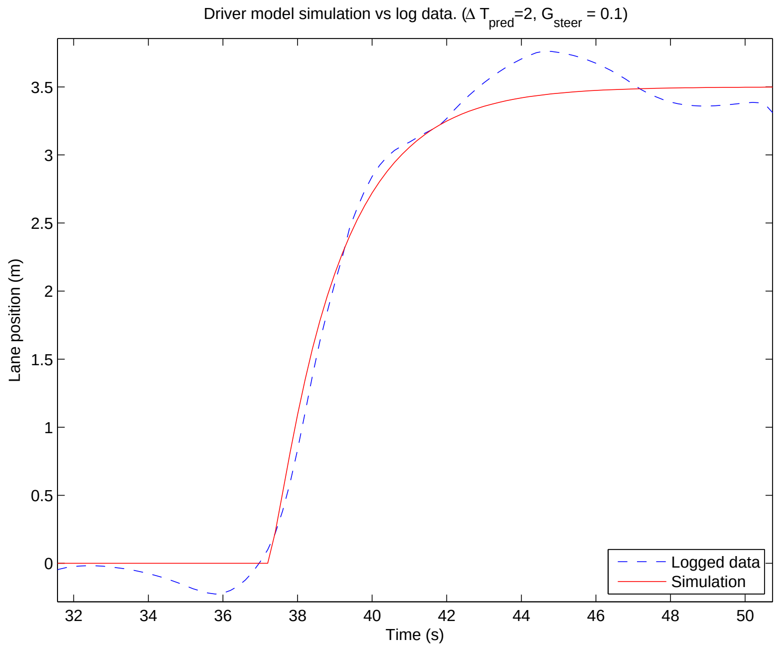

Figure 10: The figure shows the result of a step response applied to the suggested lateral control model. The step response can be seen as a simulation of a lane change i.e. the reference value suddenly changes from the source lane to the target lane. The result of the simulation is compared to a lane change from log data from the study.

Figure 10 illustrates a simulation where the above model is subject to a step in reference value, corresponding to a lange change manoeuvre. The steering wheel angle is then fed into the vehicle model described in section 5.2. The resulting profile of Lane position seem to behave rather similar to real lane change manoeuvres recorded at the the experiment in terms of rise time and lateral velocity during the lane change movement.

The results presented later on in this report shows that the model is able to describe lane change behaviour accurately enough for the present purposes.

The lane change from log data in Figure 10 shows an overshoot when centering in the new line. Such an overshoot is present in some, but not all lane changes logged in the experiment. The overshoot may suggest that the driver's ability to correctly predict lane position is a bit degraded during the lane change, maybe due to a rather high lateral velocity or because the driver is paying more attention to surrounding traffic and mirrors and less on the vehicle handling.

6.2 Tactical driver model (F_T)

On the tactical level, the driver makes decisions such as which lane to use, whether to exit or remain on the same road, whether to pass the vehicle in front or not etc. Such decisions are based on factors such as road geometry, other vehicles or obstacles in the path, destination and so on. In this experiment such factors are not observable, so the only model we can make on the tactical behaviour is a probabilistic one, considering the overall probability of lateral manoeuvres. We will assume that the intention of the driver at all times is one of a set of manoeuvres $M = \{m_1, m_2, \ldots, m_n\}$. The transition probabilities between these manoeuvres are defined by the matrix tp_{ij} where:

$$tp_{ij} = Prob(F_T(t) = m_j | F_T(t-1) = m_i) \tag{14}$$

In other words the tp_{11} is the probability that the driver will remain executing manoeuvre 1 whereas tp_{12} is the probability that the driver will switch from manoeuvre 1 to manoeuvre 2. In case infrastructural information is available tp_{ij} could be made dynamically adaptive to constraints in the surroundings. E.g. if there is only one lane in each direction, the probability of switching into a lane change would be zero. In that case any indication of movement outside the lane would rather indicate a takeover, a curve-cutting or an unintentional lane departure.

7 Classification of driving manoeuvres

The manoeuvre classification task is to estimate $F_T(t)$ i.e. the driving manoeuvre that the driver is performing at time t. In this section, a method for estimating $F_T(t)$ is described. The input to the method is the sequence of control behaviours that best explains the measurement data. The task of the classification is then to find the sequence of manoeuvres that best explains the sequence of control behaviours. It could be the case that a certain control behaviour corresponds directly to a manoeuvre, but it could also be the case that a manoeuvre is built up of several control behaviours applied in a certain order.

7.1 Estimating the operational behaviour

The observable data is the vehicle state vector $\mathbf{x(t)} = \{L(t), \delta(t), v(t), \Theta(t)\}$. Given the state vector we will let our model F_O simulate the driver control behaviour one timestep ahead, computing the estimations $\hat{\mathbf{x}}_i(t + \Delta t)$, $i = 1 \ldots n$ using each of the control behaviours in the set $B = \{b_1, b_2, \ldots, b_n\}$. When the measurement of $\mathbf{x}(t+\Delta t)$ is available the estimations $\hat{\mathbf{x}}_i$ are compared to the measurement and the control behaviour providing the best prediction is chosen as an estimation of $F_O(t)$.

$$\hat{F}_O(t) = b_i, \tag{15}$$

where

$$i = argmin_j(|\hat{\mathbf{x}}_j(t + \Delta T_{pred}) - \mathbf{x}(t + \Delta t)|) \tag{16}$$

Since only the lateral behaviour is studied no prediction of $v(t)$ is made by the model. The vehicle angle $\delta(t)$ relative to the road is computed from the $L(t)$ and $v(t)$, so comparing the model prediction of $\delta(t)$ with the measurement would be equivalent to comparing the lane position estimate with the measurement.

The steering angle $\Theta(t)$ is predicted by the driver model, but the estimate is not used to evaluate the performance of the model. The reason is that the model is not accurate enough in predicting the steering wheel angle which, as discussed previously, is very individual and depends on many inputs and considerations made by the driver. In addition the human steering wheel angle depends on the road curvature whereas the model's steering wheel angle don't consider road curvature. The result of the steering wheel angle in terms of lane position, on the other hand, turns out to contain accurate clues about the driver intention. Pentland & Liu [11] made similar observations and concluded that the steering wheel angle chosen by the driver is subject to microevents not easily captured by a model. Also they chose to evaluate only the ability of the models to predict lane position.

In other words we only compare the estimation of $L(t)$ with measurements in order to evalautet the different control behaviours. Equation 16 is thus reduced as shown in Equation 17

$$i = argmin_j(|\hat{L}_j(t + \Delta T_{pred}) - L(t + \Delta t)|) \tag{17}$$

To summarize, the control behaviour b that provides the most accurate prediction of the future vehicle lane position is assumed to be the one best describing the current control behaviour of the driver.

7.2 Estimating the tactical bahviour

Hidden Markov Models (HMM) [15] provide means for modelling a system where an unknown and unobservable underlying state gives rise to a series of observable entities.

A first order HMM assumes that the hidden state X at time t only depends on the previous state at time $t - \Delta t$, whereas in an n'th order HMM the state X depends on the state at times $t - \Delta t$, $t - 2\Delta t$, ... ,$t - n\Delta t$.

A first order HMM uses transition probabilities tp_{ij} for the event that the state X_i is followed by the state X_j. The model also uses the observation probabilities op_{kl} expressing the probability of observing y_k given that the hidden state is X_l. Note that the observation probabilities are assumed time independent, i.e.

$Prob(Y(t) = y_i | Y(t - \Delta t) = y_j) = Prob(Y(t) = y_i)$ given that the hidden state is the same at time t as at time $t - \Delta t$.

In the present experiment, the manoeuvre m being executed (the state of F_T) is represented by the hidden state of the Markov model, whereas the observed control behaviour b (the choice of F_o) is represented by the observation (see Figure 11). In other words, a manouvre m may be executed by utilizing a series of control behaviours b_1,,b_n. The manouvre is considered to be non-observable whereas the control behaviour is considered as the observable entity of the HMM.

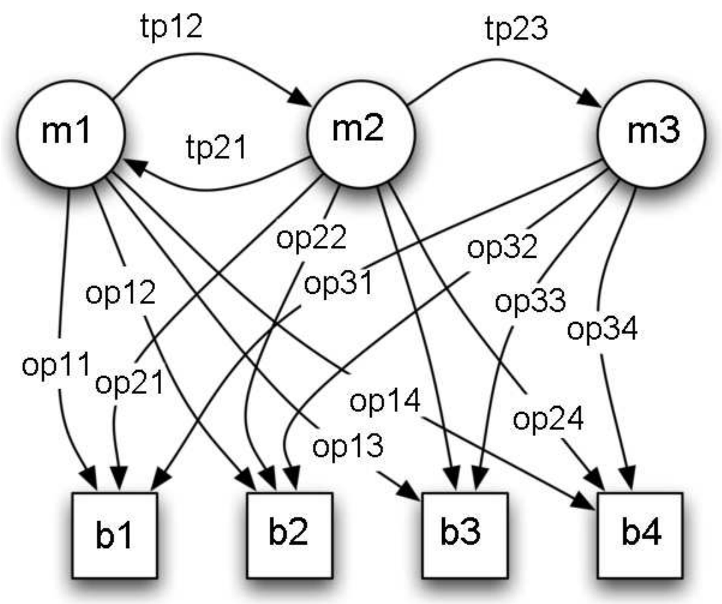

Figure 11: The figure illustrates the concept of hidden markov models applied to this study. The hidden state, illustrated by a circle, represents the manoeuvre at time m_i at time t_i. The squares illustrate the control behaviour b_i, observed at time t_i. The observation probabilities op_{kl} denotes the probability of observing the control behaviour b_l given that manoeuvre m_k is ongoing. The transition probabilities tp_{ij} denotes the probability that manoeuvre m_i is followed by manoeuvre m_j.

Given a series of observations and the transition and observation probabilities the Viterbi algorithm [18] solves the problem of finding the most probable sequence of hidden states that could have produced the series of outputs. The Viterbi algorithms only uses an approximation in that it only considers the most probable path of hidden states unlike other related algorithms as for example the forward-backward algorithm [18] that considers all possible paths. The approximation, which makes it computationally very efficient, is typically good enough for deriving the state sequence. The Viterbi algorithm as used in this experiment is provided as a Matlab script in Appendix A.

In our experiment we will consider $F_T(t)$ as the hidden state and $\hat{F}_O(t)$ as the observation. The observation probabilities will be estimated using the experiment data. The estimation $\hat{F}_T(t)$ will be computed using the Viterbi algorithm, based on the composition of the recently observed control behaviours.

Since a manouvre, according to our model, consists of the execution of different control behaviours in a certain order, it could be argued that the HMM is insufficient in that it assumes a static distribution of observations within a state. Given a certain manoeuvre, the observations have the same probability in the beginning of the manouvre as at the end of the manoeuvre and independently of which observations (control behaviours) have occurred previously during the manoeuvre. A possible extension of the HMM to take the intra-manouvre control sequence into account would be to assign transition probabilities also to the observations as illustrated in Figure 12. This approach, however has not been tested in this study.

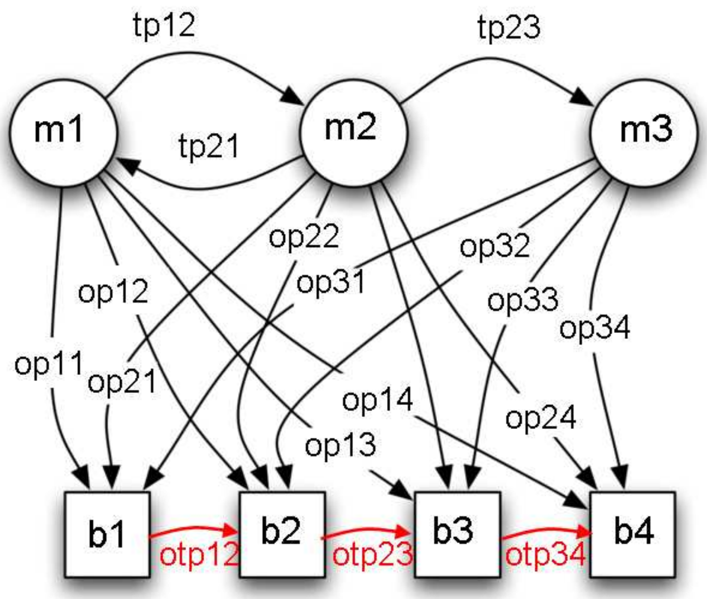

Figure 12: A possible extension of a hidden markov model considering observation transition probabilities.

8 The simulator experiment

This section describes the simulator experiment performed to develop and evaluate the manouevre prediction algorithm. The simulator data collection is described followed by the result section where the performance of the system is reported.

8.1 Data collection

In order to collect data for this experiment a fixed base simulator was used. The mockup is a Volvo S80 located in front of a 180°screen presenting the visual simulation. The simulator did not have any graphical information in the rear view mirrors, so the subjects were not able to check the mirrors when changing lanes.

Figure 13: The driving simulator used in the experiment.

Each subject drove a route including motorway, rural road and urban roads. The test route was approximately 13km long and took about 10 minutes to drive. The speed limit in the test drive was between 50-110km/h and the lane width was either 3.1 or 3.6 meters depending on the road type. The vehicle width was 1.83m.

The test subjects were instructed to drive normally and adapt their speed in the same way as they would in real life driving. At certain points along the road the subjects were instructed by the test leader to perform certain manoeuvres such as changing lanes or passing the vehicle in front. When possible, the instructions were given well before the point were the manoeuvre was to take place, for example ”You will soon catch up with a slower vehicle, pass it when suitable” or ”Change into the left lane when you find a good opportunity”. This way, the manoeuvres can be expected to be more similar to those of real life driving, since the test subject performs the usual judgements and preparing actions.

The test leader used a button to create annotation marks in the logfile to mark the occurrance of manouevres. The annotations marked the onset of the manoeuvres as judged by the testleader. In other words, the annotations mark the moment when a human observer could identify the onset of the manoeuvre.

The drawback with such a subjective annotation is that the moment of annotation may not be entirely consequent as it is up to the judgement of the test leader. The alternative would have been to give instructions of the type ”Change to the left lane now” and let the annotation mark the moment when the instruction were given. This would, on the other hand, introduce a behaviour of following orders rather than driving naturally. It was decided that the subjective annotation was the better option.

8.1.1 Result of data collection

A total of 9 test subjects were used. The profile of the subjects is summarized in Table 3 below. Most test subjects were below 30 with a driving experience of 5-10 years, driving typically a couple of times per week.

Table 3: Profile of test subjects.

Gender	3: female, 6: male
Age	7: below 30, 1: 30-40, 1: 50-60
Driving experience	1: less than 5 years, 6: 5-10 years, 1: 10-20 years, 1: 30-40 years
Driving frequency	4: once a week or less, 2: 2-3 times a week, 3: every day
Driving amount	5: less than 100km/week, 3: 101-200km/week, 1: 201-300km/week
Professional driving	There were no professional drivers.

Table 4 shows the number of occurrencies of each maneouvre in the log data.

Table 4: Occurrence of manoeuvres in logged data.

Type of manouevre	Number of occurrencies
Left lane change	39
Right lane change	40
Motorway exits	9
Takeovers	19
Braking/stopping	40
Normal driving	83min

Data about the vehicle state was collected according to Table 5. The frequency of the data logging was not entirely consequent but varied between approximately 30-60Hz following the update rate of the simulator.

Table 5: Data logged during the test drives.

Signal description	**unit**	**Symbol in this report**
Time	s	t
Road type	$enum$	-
Vehicel velocity	km/h	v
Acceleration	m/s^2	a
Yaw rate	s^{-1}	ω
Brake pedal position	$0-1$	-
Accelerator pedal position	$0-1$	-
Steering angle	$^\circ$	Θ
Lane position	m	L
Trip distance	m	-
Speed limit	km/h	-
Heading	$^\circ$	-
Position in world coordinates (X,Y,Z)	m	-
Road curve radius	m	R
Manoeuvre annotations	$0/1$	-

It was discovered that the lane position data suffered from a defect. As illustrated in Figure 14 a kind of waveform seem to be overloaded on top of the signal. This defect is caused by the fact that curved road segments in the simulator are in fact built up by peices of straight road segments. Using this knowledge, it was possible to remove the waveform almost entirely from the data. The procedure of doing so is explained in further detail in B.

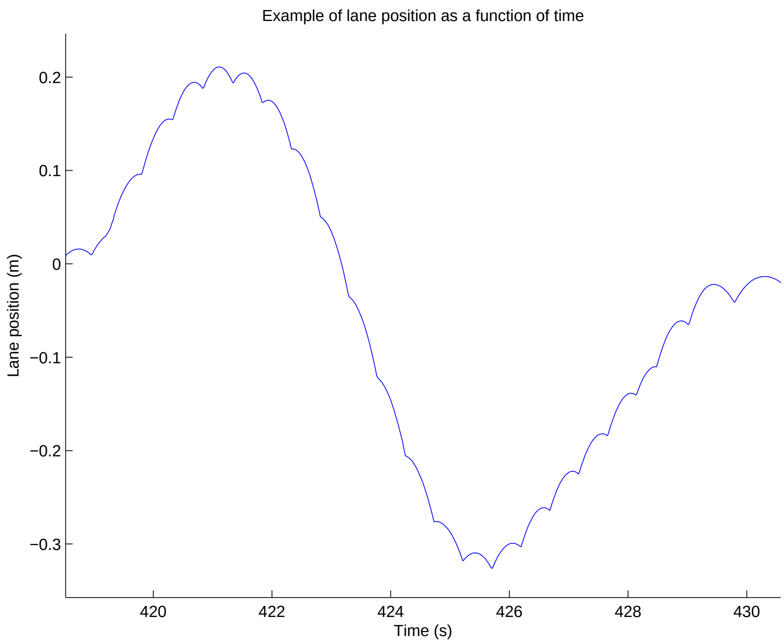

Figure 14: An example of lane position data where the overloaded waveform is visible.

In addition, there are certain parts of the test route at which the lane position data is of generally lower quality both in terms of noise and of the presence of discontinuities. These road parts are complicated road structures such as roundabouts and urban road segments with crossings. In such situations a real lane tracking sensor would normally not track the lanes in its normal manner, and it would most probably indicate that its data is of low confidence. A similar approach will be used in the analysis of the collected data.

8.2 Analysis

In this chapter, the results of applying the beforementioned manoeuvre prediction algorithm is reported. Results are listed for different versions of the algorithm and the influence of different parameter settings on the performance is investigated.

8.2.1 Data preparation

This study focused solely on lateral manoeuvres. The only manoeuvres to identify were left and right lane changes. The annotation data from the logging was modified in order to fit this purpose. Moreover, the areas mentioned in chapter 8.1.1 where the lane tracking signal is of low confidence was marked so that the algorithm may disregard from such data.

The algorithm also disregarded from data where the vehicle velocity was below a certain threshold. The reason is that the driver model seemed to perform badly at very low speeds. This may not be too surprising since the model was derived with a lane keeping task in mind typical for motorway and rural road driving where speeds are high. At very low speeds, other objectives may apply, and the predictive behaviour may not be valid as it seem to be in higher speeds. Telling from inspection of the logged data, the model seemed to perform acceptably well at speeds above 10m/s. Disregarding from lower speeds than that seems like an okay limitation to introduce, since most of the test route was driven at higher speeds than that.

To summarize, the following modifications were made to the annotation signal:

1. Overtaking manoeuvres were denoted as a lane change to the left. The lane change were considered to take place from the start anntotation until the first local maximum of lane position after entering the other lane (see Figure 15).

2. Motorway exits were denoted as right lane changes.

3. Any other manoeuvres apart from left and right lane changes were removed.

4. Manoeuvres overlapping with an area of data that was to be disregarded from either because of poor data quality or low speed were removed completely.

In fact, no manoeuvres overlapped with the regions of low data quality, but a few coincided with low speed. If the overlap of a manoeuvre with a low speed section was only partial, that manoeuvre was removed even if correctly recognized by the algorithm. This was done as a precaution not to pollute the results with situations not representable for the algorithm.

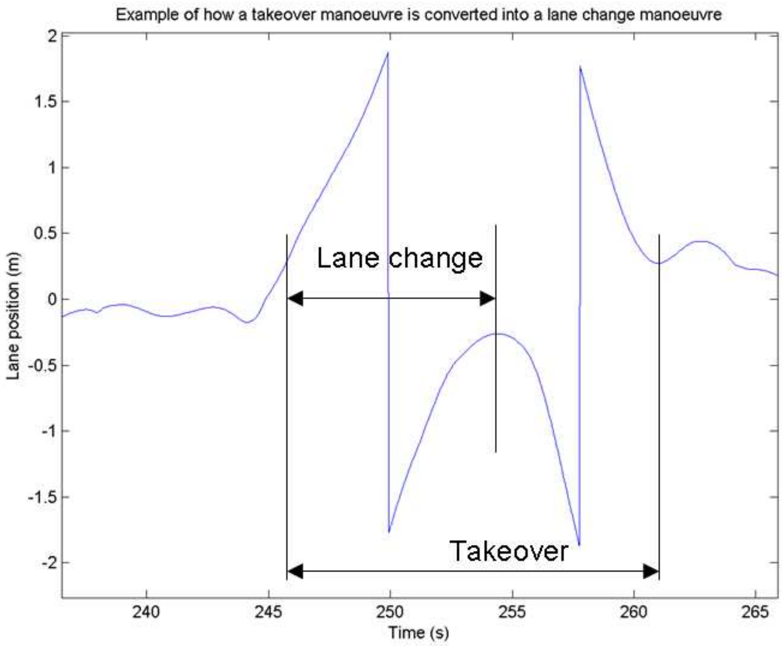

Figure 15: An example of how a takeover is changed into a left lane change with respect to the analysis.

When these modifications had been done, the amount of manoeuvres presented in Table 6 ware left to analyze.

Table 6: Occurrence of manoeuvres in logged data after modification.

Type of manouevre	Number of occurrencies
Left lane change	49
Right lane change	47
Normal driving	83minutes

8.2.2 Computation of evaluation metrics

The analysis of different configurations of the system includes calculating a set of parameters indicative to the performance of the system. The indicators used in the evaluation should capture

- the ability of the system to correctly detect driving manoeuvres
- the ability to predict manoeuvres as early as possible
- the amount of false classification made by the system both in terms of number of occurrencies and length of each occurrence.

Since the episodes of normal driving are considerably longer than those of manoeuvres, a terminology that is somewhat adapted to the present study is needed. Figure 16 illustrates possible conditions of true and false classification of the present experiment. Theoretically, there is another possible false classification in the case that a manoeuvre is classified by the system as another manoeuvre than the true one. It turned out that this never happens on the data logged in this experiment, so that failure mode can be disregarded from.

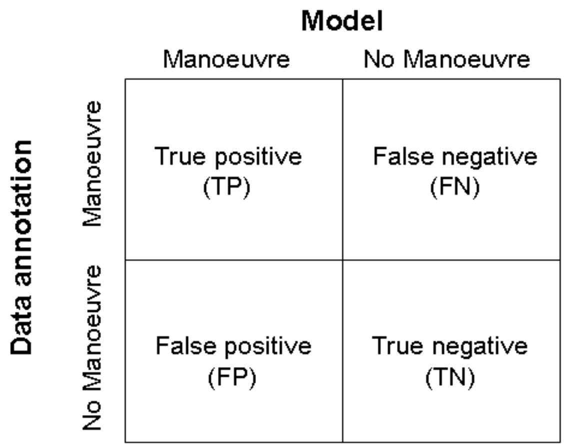

Figure 16: Possible true and false classification of the experiment.

The indicators listed in Table 7 were used for evaluating the performance of the system.

Table 7: The table lists the parameters used to evaluate the system.

Evaluation metric	Definition	Unit
Positive Hit Rate (PHR)	The proportion of manoeuvres correctly classified $\left(\frac{TP}{TP+FP}\right)$	%
Mean Lane position at detection (LD)	The mean lane offset (L) at the moment when the manoeuvre is correctly classified	m
Mean Time to line crossing at detection (TLCD)	The mean amount of time from correct classification until the wheels cross the line	s
False Occurencies (FO)	The number of occurrences of false classification $(FN + FP)$	-
Time of False Classification (TFC)	The percentage of time where a false classification (FN or FP) is made $\left(\frac{t_{FP}+t_{FN}}{t_{total}}\right)$	%

The goal of the prediction algorithm was to detect driving manoeuvres correctly as early as possible with the lowest possible amount of false classification. Early detection means low values of LD and high values of TLCD.

The median of lane position and TLC at detection was also computed but is only reported when there is a significant difference between the mean and median values, indicating that outliers have corrupted the mean value.

When calculating the results, the *leaving-one-out method* [19] was used, i.e. any system parameters estimated based on the log data is estimated using data from all the subjects except the one to be evaluated. The parameter estimation and the evaluation is repeated until all the subjects have been used for evaluation. The choice of using this method rather than using a part of the test route for training and the rest for evaluation has three reasons. The test route was poorly designed for the latter method since the road type changes several times. It would have been difficult to find a training set representative for the entire drive. It is also of particular interest to evaluate the ability of the system to function on unseen drivers, reason being that it is desirable for an in-vehicle system, particularily a safety related system, not to be dynamically adaptive. In addition, the leaving-one-out method gives the opportunity to use all data for both training and evaluation, which is good when the total amount of data is relatively small.

The process of computing the evaluation metrics goes as follows:

1. Apply the set of control behaviours on the log data for all subjects, store the result in terms of the best performing predictor at each time step $F_O(t)$.
2. Estimate the HMM parameters a_{ij} and b_{kl} based on data from all but one subjects.
3. Estimate the manoeuvres in terms of the hidden state $F_T(t)$ using the Viterbi algorithm for the test subject not included in the estimation of HMM parameters.
4. Calculate the evaluation parameters for this subject.
5. Repeat steps 2-4 for all subjects.
6. Calculate the overall results.

The estimation of a_{ij} is simply done by counting the occurrences of transtitions from maneouvre i to manoeuvre j according to the annotation signal. Similarily, b_{kl} is estimated by counting the occurrences of the observable state $F_O(t)$ given which maneouvre is going on according to the annotation signal. The matrices are normalized so that so that $\sum_j tp_{ij} = 1$ and $\sum_k op_{kl} = 1$ respectively.

8.3 Analysing parameter settings

In this section the relation between parameter settings of the system and the performance parameters are analysed. The system settings to be analysed are listed in Table 8.

Table 8: The most important system parameters whos effect is analysed below in this section.

System parameter	Definition
ΔT_{pred}	The assumed driver prediction time used in Equation 12
Δt	The timestep used when evaulating the different control behaviours in Equation 17

Assuming that the system parameters are mutually independent, they will be varied separately and their respective relation to performance and false classification will be studied.

8.3.1 Varying driver prediction time ΔT_{pred}

In the first analysis the driver prediction time ΔT_{pred} is varied in a range from $0{,}5 - 3{,}0s$ in steps of $0{,}25s$. The driver prediction time can be expected to vary individually between drivers and maybe also within subjects depending on the situation. The purpose of this test, however, is first of all to find the relation between algorithm performance and ΔT_{pred}. The aim is not to draw any conclusions about what the prediction time of the drivers actually is.

The system configuration during this test apart from the driver prediction time is summarized in Table 9. The control behaviours used are according to Equation 13 with one corresponding to following the current lane, one corresponding to a left lane change and one to a right lane change i.e. with L_{ref} set to 0, $3{,}5$ and $-3{,}5$ respectively.

In this system configuration, there is obviously a direct mapping between control behaviour and manoeuvre, i.e. the control behaviours with L_{ref} other than zero corresponds directly to left and right lane changes. In this case, one may ask what the Hidden Markov Model is good for. In fact, the HMM can be removed and the observed control behaviours can be used as indicators of the manoeuvre directly. However, if doing so, the amount of false classification increases. The HMM has the effect of filtering the classification with lower false rate as a result.

The results are shown in Figures 17 to 20.

Table 9: System configuration in test.

Parameter	**Value**
$b_1 : L_{ref}$	$0m$
$b_2 : L_{ref}$	$3{,}5m$
$b_3 : L_{ref}$	$-3{,}5m$
Δt	$0{,}2s$
G_{steer}	$0{,}1$

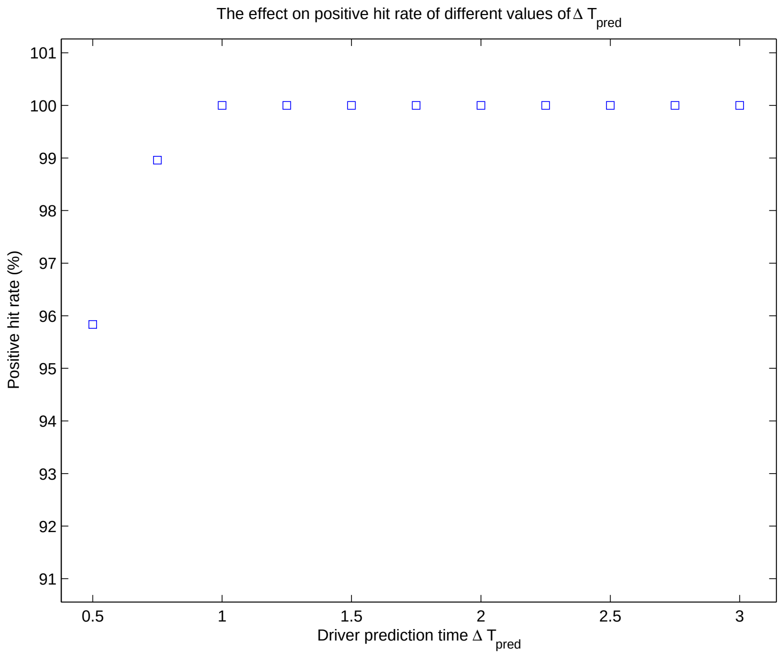

Figure 17: Positive hit rate dependence of ΔT_{pred}.

The hit rate during this experiment was 100% for all values of ΔT_{pred} larger than 1s.

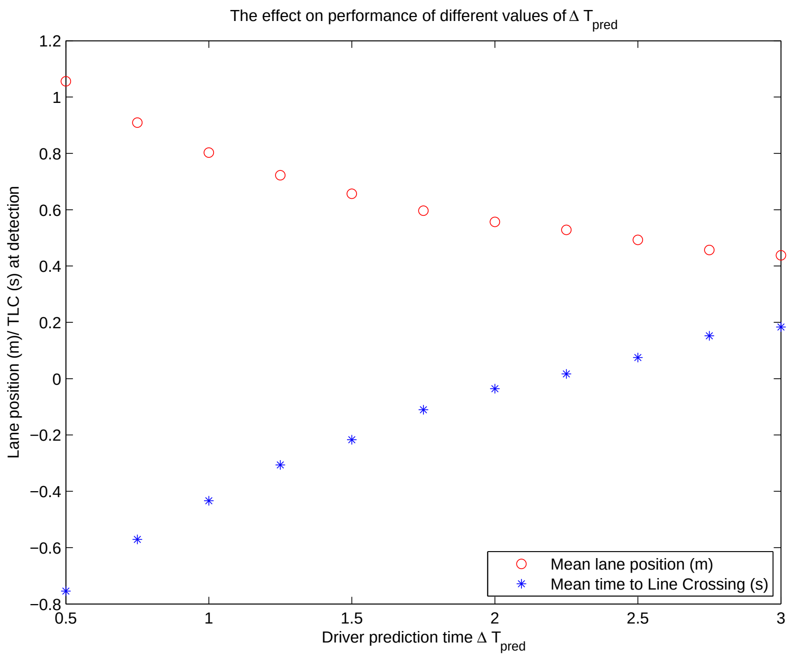

Figure 18: Dependence on lane position and TLC at detection of ΔT_{pred}.

Obviously, high values of ΔT_{pred} yields earlier detection (lower values of mean lane position and high values of mean TLC at detection). A high value of driver prediction time yields smoother driving and slower corrections and thereby reduces the difference between a staying in lane behaviour and a changing lane behaviour. Thus the algorithm can be expected to react on smaller evidence of a lane change than for lower values of driver prediction time. Reacting to small evidence, on the other hand, ought to yield more false classification.

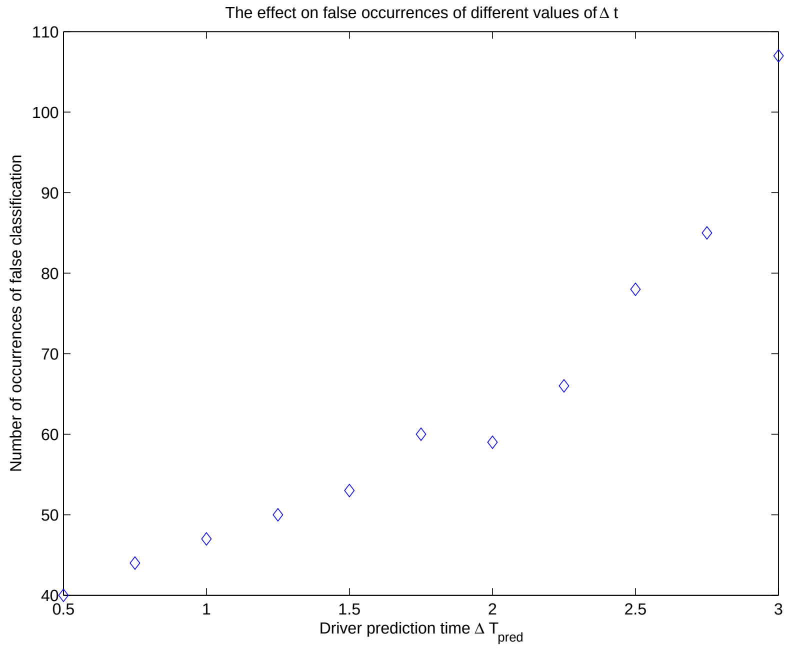

Figure 19: False occurrences dependence of ΔT_{pred}.

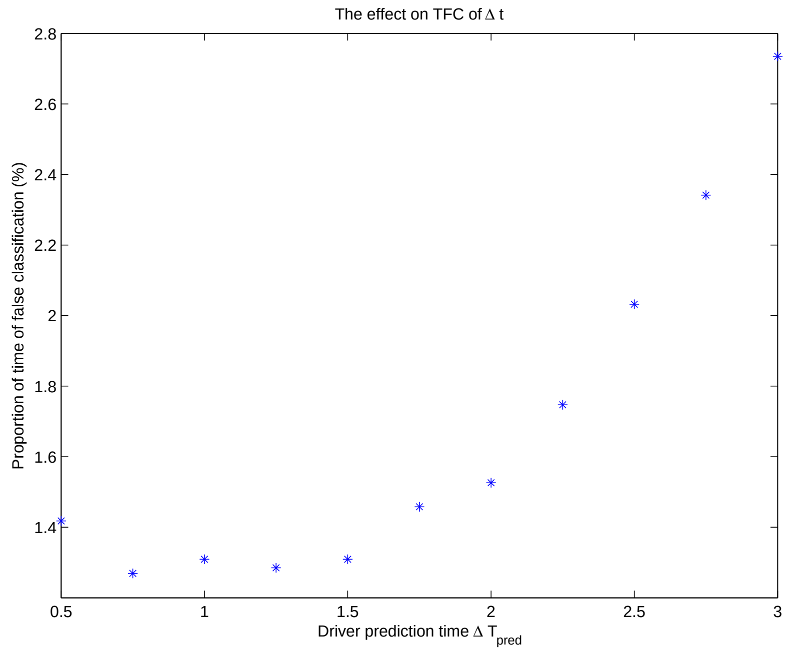

Figure 20: Time of false classificationt dependance of ΔT_{pred}.

As shown in Figures 19 and 20 the degree of false classification indeed increases with higher values of ΔT_{pred}. The number of occurrences of false classification i.e. seem to increase more or less linearly with ΔT_{pred} whereas the false amount seem to grow more exponentially, especially for values of ΔT_{pred} higher than approximately $2s$, indicating that not only the frequency but also the average length of false classifications increases with higher values of ΔT_{pred}.

8.3.2 Varying the time step Δt

In this next test, the timestep Δt used in Equation 17 is varied in order to find its relation to performance and false classificiation. A static value of ΔT_{pred} of $2s$ is chosen for this test. The time step Δt is varied from $0{,}05s$ to $0{,}3s$ in steps of $0{,}025s$.

Table 10: System configuration used in the test.

Parameter	Value
$b_1 : L_{ref}$	$0m$
$b_2 : L_{ref}$	$3{,}5m$
$b_3 : L_{ref}$	$-3{,}5m$
ΔT_{pred}	$2s$
G_{steer}	$0{,}1$

The results of this test are shown in Figures 21 to 23.
The positive hit rate during this experiment was 100% for all values of Δt.

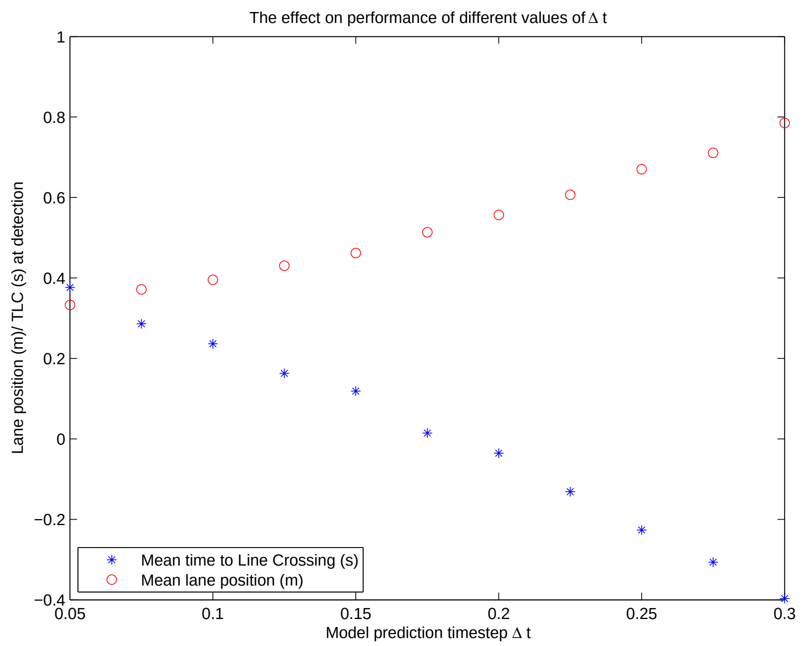

Figure 21: Performance dependance of Δt.

As shown in Figure 21 the detection of the algorithm is later (higher values of LD and lower values of TLCD) for higher values of Δt and the relation seem to be linear. This is according to expectation since a high value of the time step causes a long idle time for the algorithm between calculating the predicted vehicle state and waiting for the measurement it should be compared with. On the other hand, a longer time step serves to filter the algorithm input signals yielding lower amounts of erroneous classification.

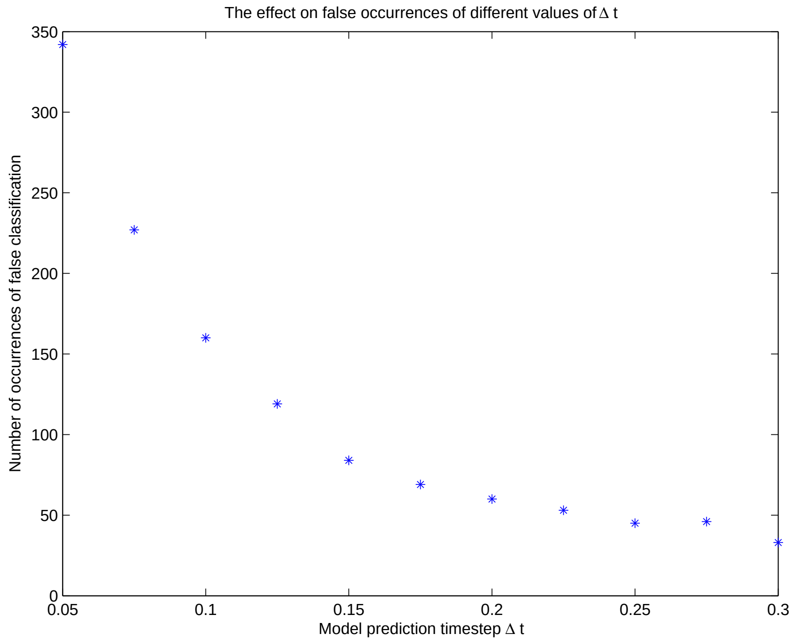

Figure 22: False occurrences dependance of Δt.

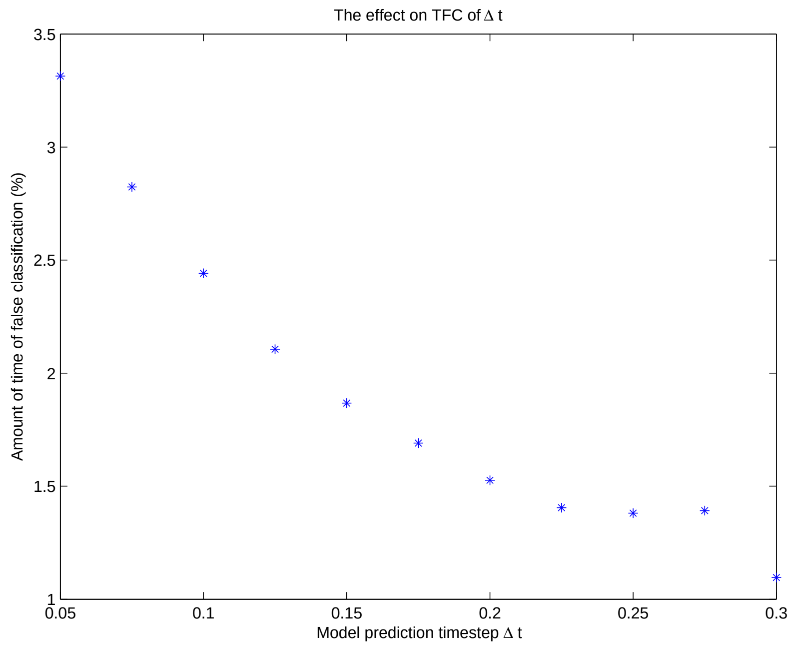

Figure 23: Time of false classificationt of Δt.

Figures 22 and 23 illustrate how the false rate and false amount decreases in a similar manner for higher values of Δt. The degree of false classification seem to drop rapidly with increasing time step until a value of about $0{,}2s$ after which the slope is considerably smaller.

8.4 Analysing the set of control behaviours

From the results from varying Δt and ΔT_{pred}, a set of values can be chosen for use in the rest of the analysis. The choice has to be a tradeoff between early detection and a low amount of faults. Given an application, parameters have to be chosen depending on the requirements of that application.

For the remainder of this report $\Delta t = 0{,}2$ and $\Delta T_{pred} = 2$ will be used as default values. Using these settings, the higher false rate is avoided without sacrificing too much of performance.

Using the above settings with the set of control behaviours used in the previous analysis yields the system configuration in Table 11 and the evaluation metrics given in Table 12. This setup will be used as reference throughout this section.

Table 11: System configuration in test.

Parameter	Value
$b_1 : L_{ref}$	$0m$
$b_2 : L_{ref}$	$3{,}5m$
$b_3 : L_{ref}$	$-3{,}5m$
ΔT_{pred}	$2s$
G_{steer}	$0{,}1$

Table 12: Evaluation metrics of the system with default settings used as reference.

Parameter	Value
Positive hit rate	100%
Mean lane position at detection	$0{,}55m$
Mean TLC at detection	$0s$ (median value is $0{,}2s$)
False occurrencies	60
Time of false classification	$1{,}5\%$

8.4.1 Adding control behaviours with small lane offsets

The set of control behaviours used in the reference system is designed to identify the drivers intention to move out of the lane or stay in the lane. Thus, it can not be able to identify the lane change until the steering towards the neighbouring lane has started. Maybe earlier signs of the preparation of a lane change can be identified.

One hypothesis about how such a preparation phase can be identified is the assumption that the driver chooses a lane position with a bit of offset towards the lane to which the driver is heading while checking the mirrors and taking the final decision to execute the manouvre.

By adding two more control behaviours with L_{ref} slightly off the road center, yet not crossing the line, we may be able to identify such behaviour. Using controllers with L_{ref} of $0{,}5$ and $-0{,}5$ (The wheel crosses the line at $L = 0{,}63m$ and $L = 0{,}89m$ for lane widths of $3{,}6m$ and $3{,}1m$ respectively) in addition to the original ones the system setup of Table 13 and the results of Table 14 are achieved.

Table 13: First system configuration in test using models with small offsets.

Parameter	Value
$b_1 : L_{ref}$	$0m$
$b_2 : L_{ref}$	$3{,}5m$
$b_3 : L_{ref}$	$-3{,}5m$
$b_4 : L_{ref}$	$0{,}5m$
$b_5 : L_{ref}$	$-0{,}5m$
ΔT_{pred}	$2s$
Δt	$0{,}2$
G_{steer}	$0{,}1$

Table 14: Evaluation metrics of the system with small offset controllers of $\pm 0{,}5m$ in addition to those of the reference system.

Parameter	Value
Positive hit rate	90%
Mean lane position at detection	$0{,}53m$
Mean TLC at detection	2,7 (median value is $0{,}2s$)
False occurrencies	48
Time of false classification	19%

Increasing the small offset to $\pm 0{,}8$ the following system setup of Table 15 and the results of Table 16 are achieved:

Table 15: Second system configuration in test using models with small offsets.

Parameter	Value
$b_1 : L_{ref}$	$0m$
$b_2 : L_{ref}$	$3{,}5m$
$b_3 : L_{ref}$	$-3{,}5m$
$b_4 : L_{ref}$	$0{,}8m$
$b_5 : L_{ref}$	$-0{,}8m$
ΔT_{pred}	$2s$
Δt	$0{,}2$
G_{steer}	$0{,}1$

Obviously, this approach do not improve the system at all. The amount of time of false classification becomes very large. The reason is probably that driving closer

Table 16: Evaluation metrics of the system with small offset controllers of $\pm 0{,}8m$ in addition to those of the reference system.

Parameter	Value
Positive hit rate	94%
Mean lane position at detection	$0{,}60m$
Mean TLC at detection	$0{,}63$ (median value is $0{,}2s$)
False occurrencies	80
Time of false classification	13%

to either line is approximately as common in lane change situations as in normal driving and the additional control behaviour only serves to confuse the detection system.

Since a major part of the test route is on rural roads with rather curvy profile, it is not surprising that drivers do not stay in the middle of the road during normal driving. The test results in Table 16 shows that the amount of false classification is as much as 13% of the entire test route. Doing the same test only on the relatively straight motorway section in the beginning of the route the corresponding percentage is only 3% indicating that the curved roads are indeed a problem when applying this approach.

The result of the configuration only on motorway follows in Table 17 below:

Table 17: Evaluation metrics of the system with small offset controllers of $\pm 0{,}8m$ applied only to the motorway section.

Parameter	Value
Positive hit rate	100%
Mean lane position at detection	$0{,}50m$
Mean TLC at detection	$-0{,}3s$ (median value is $0{,}4s$)
False occurrencies	20
Time of false classification	3%

This should be compared to the reference system in only motorway conditions:

Table 18: Evaluation metrics of the reference system on only the motorway section.

Parameter	Value
Positive hit rate	100%
Mean lane position at detection	$0{,}63m$
Mean TLC at detection	$-0{,}2s$ (median value is $0{,}3s$)
False occurrencies	3
Time of false classification	1%

It seems that like the present configuration on average provides slightly earlier detection on the motorway section than the reference system. The corresponding increase in false amount indicates that the reason is that the model learns that the small offset control behaviour corresponds directly to a lane change. Since the small offset behaviour is in between the driving straight and the respective changing lanes behaviour it is not surprising that it shows up slightly before the lane change behaviour. Anyway, the small offset approach is not an improvement of the system even on motorway. The reference system can be pushed to similar results in mean lane position and TLC without exceeding the 3% time of false classification achieved using the small offset approach.

There seem to be no evidence that lane changes in the general case are preceeded by any static behaviour of driving closer to either line.

8.4.2 Using driver behaviours with different values of driver prediction time

So far we have been using the same driver prediction time for both the stay-in-lane control bahaviour and the change-lane control behaviour. A reasonable assumption may be that a longer prediction time can be applied to the lane change behaviour since the distance to the neighbouring lane is larger than that of the source lane at the initiation of the manoeuvre. For example the driver strives to steer towards the lane center within 2 seconds when driving straight but may be satisfied with ending up in the adjecent lane after, say, 3 seconds. Trying such a setup yields the configuration of Table 19 and the results of Table 20

Table 19: System configuration testing other values on driver prediction time in the lane change control behaviours than in the driving straight behaviours.

Parameter	Value
$b_1 : L_{ref}$	$0m$
$b_2 : L_{ref}$	$3{,}5m$
$b_3 : L_{ref}$	$-3{,}5m$
ΔT_{pred}	$2s$
$b_2 : \Delta T_{pred}$	$3s$
$b_3 : \Delta T_{pred}$	$3s$
Δt	$0{,}2$
G_{steer}	$0{,}1$

This approach causes the lane change identification to occur earlier but again with larger amounts of time of false classification as a consequence although the false amount is considerably lower than if applying a driver prediction time of 3s on the driving straight controller as well. Increasing the step size Δt to $0{,}25$ reduces the false amount to a similar level as that of the reference model, but then the advantage of earlier detection has vanished and the performance is lower than that of the reference model.

Table 20: Evaluation metrics of a system using longer driving prediction time for the lane change behaviour than for the driving straight behaviour.

Parameter	**Value**
Positive hit rate	100%
Mean lane position at detection	$0{,}49m$
Mean TLC at detection	$-0{,}27s$ (median value is $0{,}4s$)
False occurrencies	76
Time of false classification	$1{,}9\%$

9 Results

In this section the results of the evaluation of the maneouvre detection system is reported. Comparisons are made to the study perfomed by Pentland and Liu [11] who used a similar modelling approach.

In the previous section, a reference system was defined and compared to a couple of variants. As it turned out, the setup chosen as reference also seemed to be the one providing the best performance. Since there is no optimal solution to the choice of parameters, the reference system was chosen rather subjetively so as to avoid the highest levels of false rate without compromising performance too much. The scores of this model are listed in Table 21 below.

Table 21: Evaluation metrics of the best performing system configuration found in the analysis.

Parameter	**Value**
Positive hit rate	100%
Mean lane position at detection	$0{,}55m$
Mean TLC at detection	$0s$ (median value is $0{,}2s$)
False occurrencies	60
Time of false classification	$1{,}5\%$

The hit rate was 100% almost independently of parameter settings. In that sence the algorithm seems to be very reliable. However, if the information of the detection system is to be of any value, the detection need to preceed the crossing of the lane as much as possible. The system should detect the lane change as long before the crossing of the lane marker as possible.

9.1 Joint results of left and right lane changes

The lane position at the detection of the lane change was on average 0.55m with a standard deviation of 0.39m. At normal driving the mean and standard deviation of lane position was $-0{,}08m \pm 0{,}43m$. The mean lane position at detection is thus just outside the standard deviation of the distribution of normal driving. However, a great deal of detections do fall within the distribution of normal driving.

The lateral velocity i.e. the time derivative of lane position was on average $0{,}62 \pm 0{,}26 m/s$ at the detection of the lateral manoeuvres compared to normal driving where it was $0 \pm 0{,}7 m/s$. Thus, the values at detection are not statistically outside the distribution of normal driving indicating that the algorithm has the ability to actually recognize lane change behaviour rather than just reacting to abnormal values of lateral veloctiy.

Figure 24 shows boxplots of lane postition at detection and of lane position in normal driving. Figure 25 also shows lane postition at detection but plotted on an image of the road to illustrate the relation to the size of the road and the vehicle. Figure 26 shows a boxplot of TLC at detection.

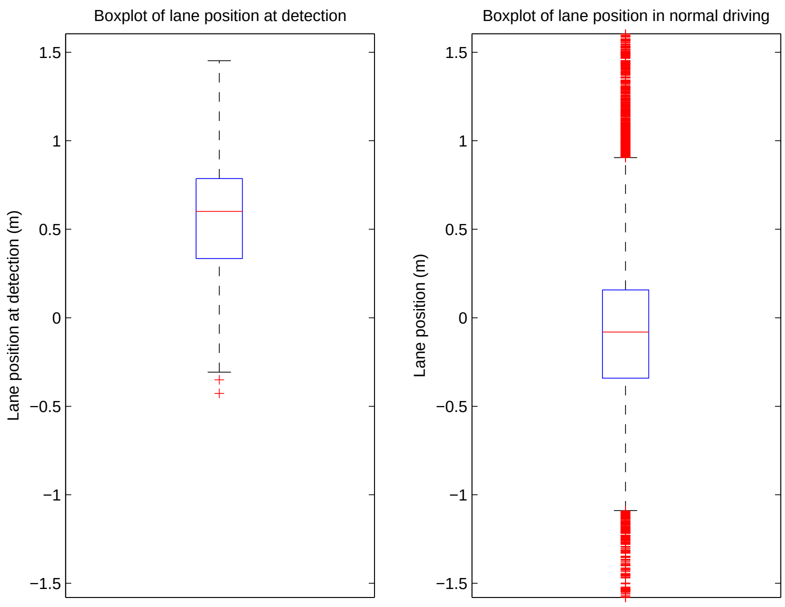

Figure 24: The boxplot to the left illustrates lane position at detection of a lane change. For comparison, the boxplot to the right shows the distribution of lane position at normal driving. The boxes do not overlap, but the lane position at detection is generally far from being an outlier according to the normal driving distribution.

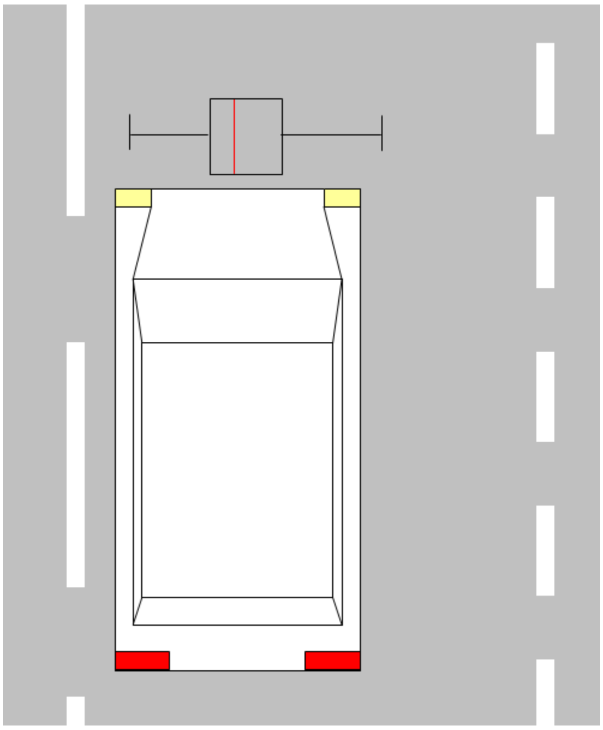

Figure 25: Boxplot of lane position at detection in relation to the size of the road and the vehicle. The boxplot is the same as the left one in Figure 24 but drawn in an image of the road and the vehicle. The lane position of the vehicle in the figure represents that of the median lane position at detection. The figure provides an indication of where in lane the vehicle generally is when lane changes are predicted.

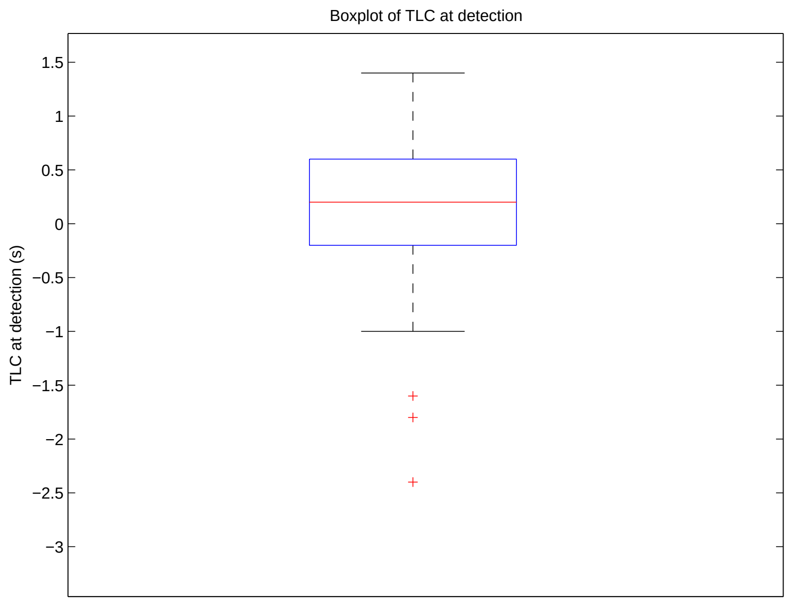

Figure 26: Boxplot of time to line crossing at detection.

In the study of Pentland and Liu [11], the mean lane position at detection was $0{,}8 \pm 1{,}38m$ meters. Their lanes, on the other hand, were 4 meters wide (ours were either 3.1 or 3.6m), so their results are not directly comparable to ours. However, if nomalizing to a lane width of 2 (i.e. so that a lane offset of ± 1 corresponds to the centre of the vehicle being right above the line), Pentland and Liu's results are $0{,}4 \pm 0{,}69m$ compared to our results which are $0{,}34 \pm 0{,}23m$.

The amount of false classification in Pentland and Lius study is not mentioned in their paper [11].

9.2 Separate results from left and right lane changes

If we analyse the results of left and right lane changes separately it turns out that left lane changes are detected considerably earlier than right lane changes.

The mean lane position at detection for left lane changes is $0{,}38 \pm 0{,}38m$ while that of right lane changes is $0{,}73 \pm 0{,}31m$. One possible explanation for this is that the test subjects were at several times instructed to change lanes to the left and then to the right, causing them to perform the right lane change too early before having centered in the left lane. This way, the right lane changes were initiated when the lane offset were already quite biased towards the target lane yielding poor values of our evaluation parameters. However it can not be totally excluded that left lane changes are in fact different from right lane changes. Figure 27 shows boxplots of lane position at detection for the two cases. It is obvious from the boxplots that there is a considerable difference in how early the algorithm is able to detect lane changes in the left and right cases.

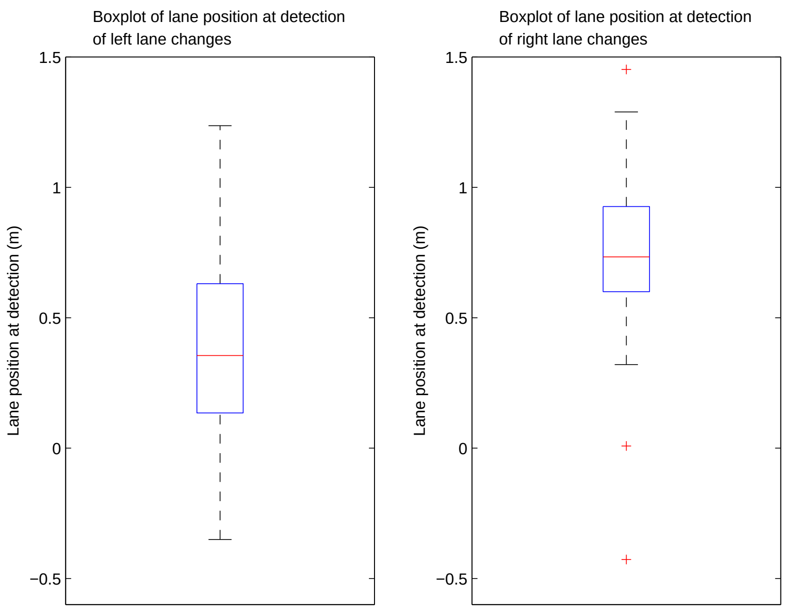

Figure 27: Boxplots of lane position at detection in the case of left and right lane changes respectively.

9.3 False detections

Using the proposed algorithm, false classification occured approximately 1.5% of the time of driving. By inspecting the occasions when the algorithm made false classifications most of them appear to be one of three kinds:

1. False positives when the driver is cutting a curve on the rural road. As the algorithm is disregarding from road curvature and no modelling of trajectory planning is made, this kind of error is likely to occur.

2. False positives after a lane change. The reason behind these errors may be that the driver is unused to the simulator vehicle and sways a bit in conjunction with the manoeuvre.

3. False negatives during manoeuvres where the manoeuvre is first correctly classified than classified as normal driving for a short while, then classified correctly again. The type 2 falses, however, were few and corresponded to only 0.15% of the driving time.

Errors due to noisy or poor data were rare as can be expected in a simulator study. In real conditions in a test vehicle however, the issue is expected to be much more of a problem.

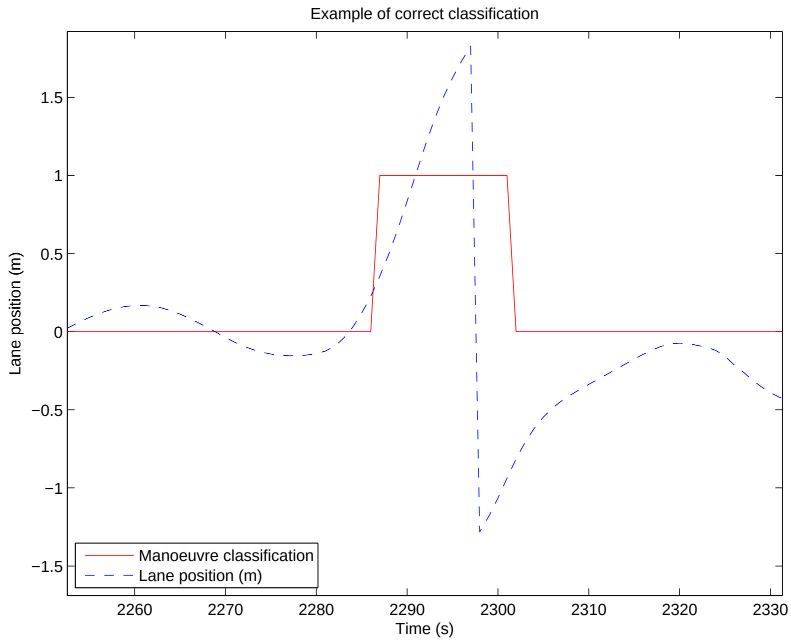

Figure 28: Example of correct classification as a left lane change.

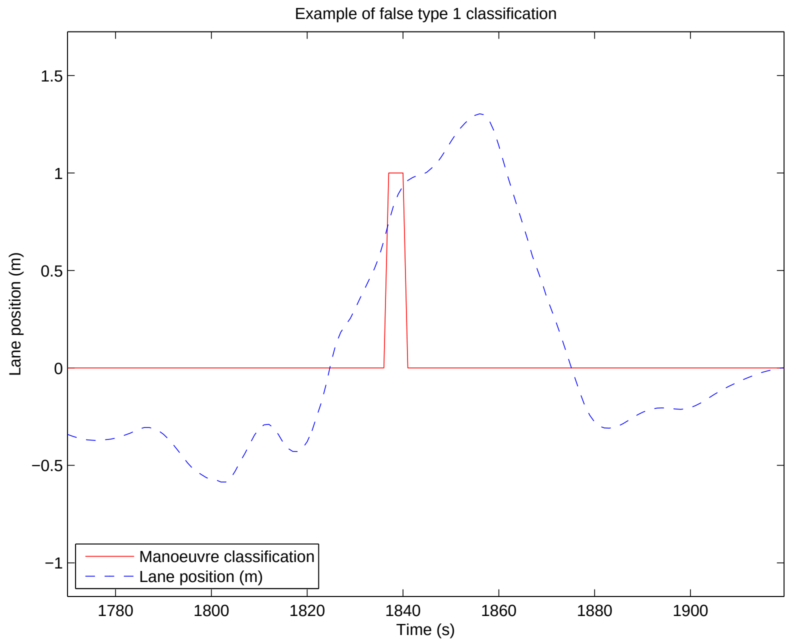

Figure 29: Example of a false positive when the driver cuts a curve on a rural road.

10 Discussion

The experiment showed that the algorithm was able to detect left and right lane changes in an early enough state for lateral position and motion to be statistically not very different from that of normal driving. The detection turned out to be considerably earlier for left lane changes than for right lane changes. Whether this was because of poor instructions to the test subjects or because there is an actual difference in left and right lane change behaivour is unknown. Since the visiblity to the right is lower than it is to the left it may be the case that right lane changes are carried out more carefully or different in some other way.

The algorithm had on average at least as good performance as that of Pentland and Liu [11] which is the one in litterature most similar to the one reported here.

The algorithm turned out to be sensitive to cutting-curves behaviour on rural roads which is not exactly a lane change situation. By use of more advanced algorithms including curvature and taking into account the trajectory planning of the driver, such situations may be better understood.

The algorithm was not applicable in speeds below 10m/s. The reason is that the driver model was insufficient at low speeds. Keeping in mind that the driver wants to keep yaw rate under a certain level in order to keep the ride comfortable the steering wheel angle can be expected to be strongly related to velocity. This applies especially at low speeds, a behaviour that was not captured by the steering model used here.

The experiment showed that the model was most successful for driver prediction times of 2s or more, which is consistent with for example [14] claiming that the prediction time is usually between 2-4 seconds.

Altering the time step used in the evaluation of model predictions the algorithm can be tuned to perform earlier detection at the cost of more false detections or the other way around. There is no optimal choice. Instead the choice depends on which application to use the algorithm for.

The algorithm is computationally very efficient in terms of both memory and processing power. In terms of memory, the algorithm needs no historical memory of any states more than one timestep behind so the total amount of memory needed can be estimated to a few hundred bytes. In terms of processing power, the entire analysis in non-optimized matlab/simulink environment on a PC takes less than 20s for 83 minutes of driving i.e. under those conditions the algorithm performs 250 times as fast as realtime. The algorithm needs no individual adaptation to the driver. In this experiment the algorithm performed well on all subjects although no form of algorithm training was made on the subject being evaluated. These factors makes it suitable for integration into a vehicle product.

The usability of the algorithm in vehicle path prediction can be expected to be high. For example, a path prediction system could hold several paths, corresponding to different manoeuvres, as probable and thereby not be as sensitive to false classification. Using the Viterbi algorithm, the probabilities of all the manoeuvres can be computed, enabling more advanced methods for choosing which one to use.

Whether the algorithm can be seen as expressing driver intent was not investigated. The data collected contained no unintentional lane departures as would have been necessary in order to see if the algorithm was able to separate intentional lane changes from unintentional ones caused by e.g. the driver falling asleep or being distracted by something.

For lane change support systems, the algorithm may very well be usable for adapting interventions or warnings according to the predicted manoeuvre.

11 Future work

The following work can be recommended as a continuation to this study:

- Apply the algorithm to real data from a test vehicle. This is interesting both in order to see if the algorithm is good enough to handle data from real sensors and to see whether the driver behaviour is different in real conditions.
- Apply more detailed driver models in order to cope with situations such as cutting curves etc. Maybe more information about upcoming manoeuvres can be extracted if better models are applied.
- Apply the same kind of modelling to longitudinal control, especially using sensor data about the position and motion of surrounding vehicles. Analysing the driver behaviour in relation to other vehicles may be helpful in understanding both lateral and longitudinal behaviour.
- Use e-horizon data in order to understand the current road configuration. The transition probability matrix of the Hidden Markov Model could be dynamically altered to reflect the availability of e.g. adjecent lanes.
- Use information about the visual behaviour of the driver in order to detect e.g. mirror glanses etc. and connect this to the driving manoeuvre modelling. The visual behaviour may be the most promising source of information to detect early preparation of manoeuvres and to really predict manoeuvres in a ”mind-reading” manner (see e.g. [10])

Acknowledgements

I would like to thank my mentor in this study, Johan Engström, as well as Trent Victor, Petter Birkedal, Emma Johansson and other collegues at Volvo Technology for teaching me and discussing issues related to this thesis. Many thanks also to Dennis Saluäär, Per Nordqvist and Emil Knabe for setting up the simulator environment and supporting me in using it.

Last but by no means least, I'm highly grateful to my friends, family and former fellow students for motivating me to finally finish this thesis.

References

[1] Andersson, Ersson: *Modelling and Implementation of a Dynamic Vehicle Model for Simulation in Real Time*, Department of Machine and Vehicle construction, University of Chalmers.

[2] Boer: *Behavioral Entropy as a Measure for Driving Performance* Wingcast

[3] Welch, Bishop: *An Introduction to the Kalman Filter*, TR 95-041, University of North Carolina.

[4] Gibson J. J.: *Visually controlled locomotion and visual orientation in animals*, British Journal of Psychology, 49, 1958, pp. 182-194.

[5] Hess, Modjtahedzadeh: *A Model of Driver Steering Control Behaviour for Use in Assessing Vehicle Handling Qualities* Transactions of the ASME, vol 115 september 1993 pages 456–646.

[6] MacAdam: *Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving* IEEE trans. on systems, man, and cybernetics, Vol. SMC-11 No. 6, 1981.

[7] McRuer, Weir: *Theory of manual Vehicular Control* Ergonomics, vol. 12, No. 4, 1969 pages 599–633.

[8] McRuer: *Personal correspondance between Erwin McRuer and Dennis Saulaar* October, 2000.

[9] Michon, J.A.: *Dealing with danger. Summary report of a workshop in the Traffic Research Centre, State University, Groningen.* (1979)

[10] Oliver N., Pentlnd A.: *Driver Behavior Recognition and Prediction in a SmartCar*, Proceedings of SPIE Aerosense 2000

[11] Pentland, Liu: *Modeling and Prediction of Human Behaviour*, Neural Computation, 11, (1999), 229-242.

[12] Pilutti, Ulsoy: *Identification of Driver State for Lane-Keeping Tasks*, IEEE trans. on systems man and cybernetics – Part A: Systems and humans vol.29, No 5, September 1999.

[13] Pilutti, Ulsoy: *Identification of Driver State for Lane Keeping Tasks* IEEE Transactions on Man, Machine and Cybernetics-Part A Systems and Humans, (29), 5.

[14] Günter Prokop: *Modelling Human Vehicle Driving by Model Predictive Online Optimization*, Vehicle System Dynamics 2001: Vol 35, No. 1, pp. 19-53.

[15] Rabiner: *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition* Proceedings of the IEEE, Vol. 77, NO.2 Feb 1989

[16] Salvucci, D. D. ; Gray, Rob.: *A two-point visual control model of steering*, 2004

[17] Tijerina et al.: *A Preliminary Assessmnet of Algorithms for Drowsy and Inattentive Driver Detection on the Road* DOT Report: HS808

[18] Andrew J. Viterbi.: *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Transactions on Information Theory 13(2):260-269, April 1967.

[19] Wikipedia: *Article on Cross-validation*, http://en.wikipedia.org/wiki/Cross-validation.

[20] Zhang: *Extended Kalman Filter*,http://www-sop.inria.fr/robotvis/personnel/zzhang/Publis/Tutorial-Estim/node16.html# SECTION00092000000000000000, Institut National de Recherche En Informatique et en Automatique.

A Viterbi algorithm script

```
v_prob = [];
v_path = [];

States = [0 1 2];

T(1,:)=[sp(1) 0 sp(1)];
T(2,:)=[sp(2) 1 sp(2)];
T(3,:)=[sp(3) 2 sp(3)];
%v_path = [;

for n=1:length(Classification)
    U=zeros(3);
    for i=States %loop next state
        Total  =0;
        argmax =0;
        valmax =0;
        for j=States %Loop source state
            prob=T(j+1,1);
            v_prob=T(j+1,3);
            p=tp(j+1,i+1)*op(Classification(n)+1,j+1);
            prob=prob*p;
            v_prob=v_prob*p;
            Total=Total+prob;
            if v_prob>valmax
                argmax=i;
                valmax=v_prob;
            end
        end
        U(i+1,:)=[Total argmax valmax];

        MaxPhi=0;
        ArgPhi(i+1)=1;
        for m=States %Loop over source states
            Phi = T(m+1,1)*tp(m+1,i+1);
            if Phi>MaxPhi
               MaxPhi=Phi;
               ArgPhi(i+1)=m;
            end
        end
    end
    [Value MostProbNextState]=max(U(:,end));
    v_path=[v_path; ArgPhi(MostProbNextState)];
    T=U;
end
```

B Removal of defects in lane position data

In order to remove the defect in lane position data a corrective function need to be computed that can be added to the original lane position signal. Following the assumption that the defect is because of the approximation of a curve as a series of peicewise straight road-sections the corrective function for each segment is given by Equations 18 and 19 below. See Figure 30 for explanations of the variables.

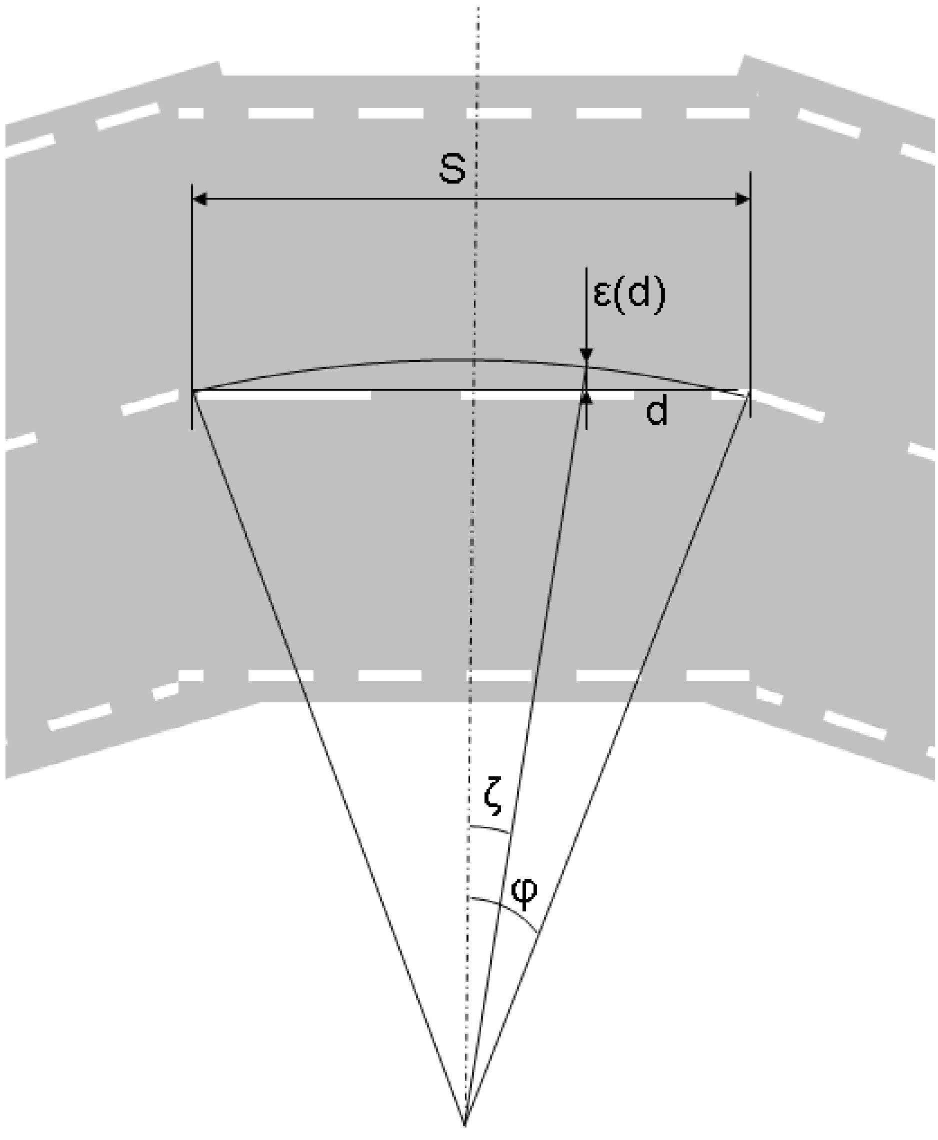

Figure 30: In order to calculate the corrective function $\epsilon(d)$ for the distance d into the segment the illustrated model is used.

The corrective function $\epsilon(d)$ for the distance d into the segment represents the distance between the centre of the road segment (which the lane positioning uses as reference) and an imagined circular centreline which is the one that we want our lane positioning measurement to refer to.

$$\epsilon(d) = R(cos\zeta - cos\phi(d)) \tag{18}$$

or in terms of the distance d and the segment length S

$$\epsilon(d) = R\left(\sqrt{1 - \frac{(\frac{S}{2} - d)^2}{R^2}} - \sqrt{1 - \frac{(\frac{S}{2})^2}{R^2}}\right) \tag{19}$$

In order to compute the corrective function for the entire signal, we need to know where the intersections between road segments are located. To do this we need to find the pairs of data points that have such an intersection in between them. Two methods were used to find these data points.

1. Finding the points in data where the road curvature R changes. Clerly there must be a new segment if the radius changes.
2. Where the radius is constant the intersections can be found by looking for peaks in the second derivative of the lane position signal.

Once the data point pairs are found, we know that the intersection is somewhere between these points. To establish the an estimate of the exact position the orignal lane position signal is extrapolated using straight lines from the two data points before and after the intersection respectively. The point where the two lines meet are used as estimation of the segment intersection point.

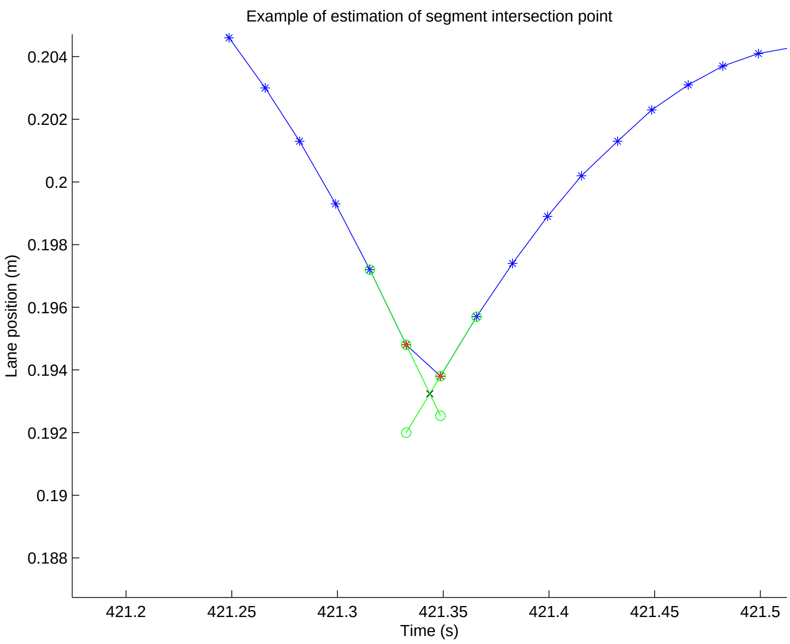

Figure 31: In the estimation of the segment intersection points the orignal lane position signal is extrapolated towards the intersection point from both directions. The intersection between the extrapolated lines is used as an estimate.

After applying the corrective action to the signal it looks as illustrated in Figure 32 below.

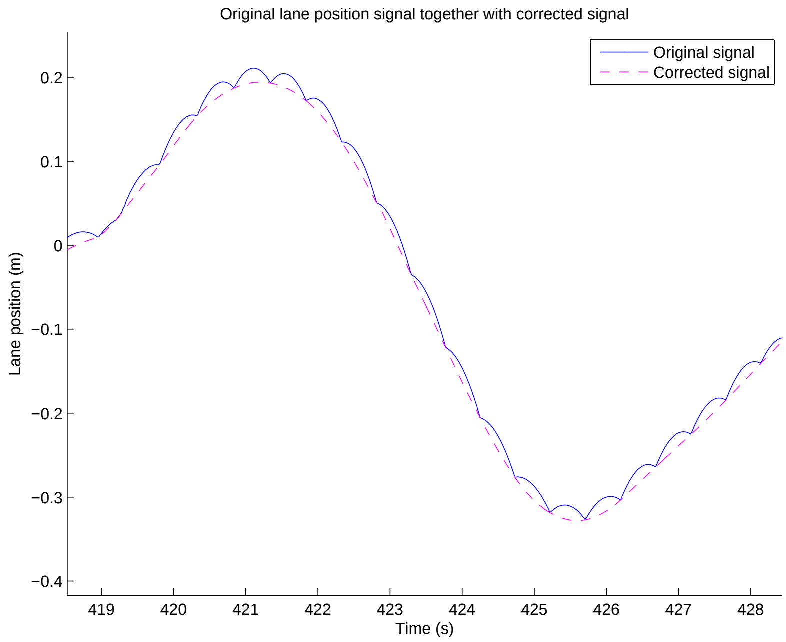

Figure 32: Comparison of original data with corrected data.