CHALMERS | UNIVERSITY OF GOTHENBURG

PREPRINT 2013:19

Detection of the Curves based on Lateral Acceleration using Hidden Markov Models

ROZA MAGHSOOD
PÄR JOHANNESSON

Department of Mathematical Sciences
Division of Mathematical Statistics
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg Sweden 2013

Preprint 2013:19

Detection of the Curves based on Lateral Acceleration using Hidden Markov Models

Roza Maghsood, Pär Johannesson

Department of Mathematical Sciences
Division of Mathematical Statistics
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg, Sweden

Gothenburg, October 2013

Preprint 2013:19
ISSN 1652-9715

Matematiska vetenskaper
Göteborg 2013

Detection of the Curves based on Lateral Acceleration using Hidden Markov Models

Roza Maghsood, Pär Johannesson
Chalmers University of Technology
SP Technical Research Institute of Sweden

October 18, 2013

Contents

Abstract

In vehicle design it is desirable to model the loads by describing the load environment, the customer usage and the vehicle dynamics. In this study a method will be proposed for detection of curves using a lateral acceleration signal. The method is based on Hidden Markov Models (HMMs) which are probabilistic models that can be used to recognize patterns in time series data. In an HMM, 'hidden' refers to a Markov chain where the states are not observable, however what can be observed is a sequence of data where each observation is a random variable whose distribution depends on the current hidden state. The idea here is to consider the current driving event as the hidden state and the lateral acceleration signal as the observed sequence. Examples of curve detection are presented for both simulated and measured data. The classification results indicate that the method can recognize left and right turns with small misclassification errors.

1 Introduction

For fatigue design the loads need to be assessed. One approach is to describe the load environment and the customer usage, which together with the vehicle dynamics define the load conditions. The characteristics of driving events used for describing customer usage can be defined using measurements obtained from specially equipped vehicles on a test track. On the other hand, measuring on vehicles in service is dificult and expensive and in general there is no access to measurements dedicated to durability. Thus, for on-board logging of events we need to use the information which is available for all vehicles by means of CAN (Controller Area Network) bus data.

A data set is available from Volvo Trucks, and the important events have been defined based on dedicated test track measurements. The problem is to identify the events from CAN-data and their frequencies. In this study we propose a method using **Hidden Markov Models** (HMMs) to detect curves based on a lateral acceleration signal. The idea is to use the driving events, i.e. straights and curves, as the hidden states and construct a HMM based on them.

The HMMs have been widely used in signal processing to recognize the events and also to predict them in the future, see e.g. the overview by Rabiner [8] with applications to speech recognition. Mitrović [6, 7] and Berndt and Dietmayer [2] used HMMs to detect driving events. They constructed one HMM for each type of driving event such as left and right curves, left, right and straight on roundabout. They created a training set by identifying events manually to build the models and evaluate them. Then for a new observation sequence, they computed the observation likelihoods based on all models and chose the driving event type with respect to the highest likelihoods.

The parameters in an HMM are the transition probability matrix, the emission matrix and the initial state distribution. They must be estimated to characterize the model. In our suggested method, we have used a single HMM for describing all events instead of constructing several different models where each HMM describes a single event. It should be more simple to estimate the parameters of one model than a lots of parameters of different models.

In an HMM, a training set is used to estimate the parameters of the model, while a test set is used to validate the model. A training set consists of all necessary information for estimating the model parameters. In our study, the training set contains all history about the curves such as the start and stop points of them. We have simulated different lateral acceleration signals with different lengths and different number of curves (events) to have some controlled training and test sets.

In Section 2 we describe the concept of HMMs and present two methods for detection of curves. For method 1 the parameters of the HMM is estimated from the training set, while for method 2 the transition matrix is re-estimated based on the test set. Examples and their results for simulated and measured data are shown in Section 3. Conclusions are presented in Section 4.

2 Hidden Markov models

Hidden Markov models are probabilistic model that can be used for detection of patterns or events in a signal. The setup is that there are two processes. The interesting process Z_t that describes the events is not possible to measure. It is thus called hidden and modelled as a Markov chain. However, what can be observed is a process Y_t whose statistical properties depends on the value of Z_t. The problem at hand is to estimate the parameters of the HMM. Based on an observation of Y_t, it is then possible to reconstruct the most probable hidden process and identify events.

In this study, three events **Right turn** (RT), **Left turn** (LT) and **Straight forward** (SF) have been considered. The idea is that one can see these three events as three hidden states and construct the HMM based on them. Figure 1 illustrates three hidden states and a sequence of observations that can be generated based on the probability distribution of observation symbols.

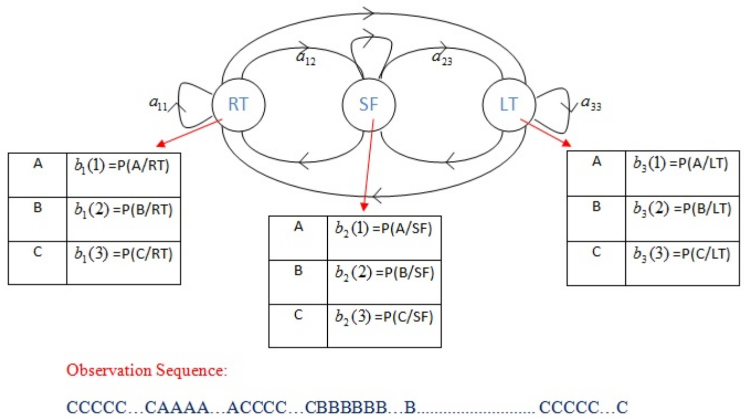

Figure 1: The hidden state sequence is modelled by a Markov chain and the observation sequence is modelled by the emission probabilities.

Let $\{Z_t\}_{t=1}^{\infty}$ be a Markov chain where Z_t denotes a hidden state at time t and has possible values $S = \{S_1, S_2, ..., S_N\}$. The transition probabilities between the hidden states are defined by the matrix $A = \{a_{ij}\}$, called transition matrix, where

$$a_{ij} = P(Z_{t+1} = S_j | Z_t = S_i),\ i, j = 1, 2, ..., N$$

and $\sum_{j=1}^{N} a_{ij} = 1$.

Further, there is another process $\{Y_t\}_{t=1}^{\infty}$ where Y_t denoting the observation symbol at time t. The sequence of observation has possible values $V = \{V_1, V_2, ..., V_M\}$ and it is observable for us. The probability distribution of observation symbols in each state is given by the emission matrix, $B = \{b_j(V_k)\}$, where

$$b_j(V_k) = P(Y_t = V_k | Z_t = S_j),\ k = 1, 2, ..., M$$

and $\sum_{j=1}^{N} b_j(V_k) = 1$.

The state where the hidden process will start is modelled by the initial state probabilities that are denoted by $\pi = \{\pi_1, \pi_2, ..., \pi_N\}$ where

$$\pi_i = P(Z_1 = S_i),\ i = 1, 2, ..., N$$

and $\sum_{i=1}^{N} \pi_i = 1$.

It has been demonstrated that a discrete HMM can be good in pattern recognition, see Rabiner [8]. We have also used a discrete HMM $\lambda = (A, B, \pi)$, where λ represents model parameters which contain the transition matrix, the emission matrix and the initial state distribution.

As mentioned before, we have three hidden states $S = \{\text{RT, SF, LT}\}$ denoting the three events right turn, straight forward and left turn, respectively. In order to estimate the parameters of the HMM, we have used the lateral acceleration signal where we also have an observation of the hidden process Z_t. This will be our training data that contains observation of both the Y-process and the hidden Z-process. We have considered lateral acceleration values as our data and thus we need to translate this continuous feature into predefined classes. Here, three classes will be used, $V = \{\text{A, B, C}\}$, that are defined as follows:

- $\text{A} = \{\text{"}\textit{lateral acceleration}\text{"} < -0.2\ m/s^2\}$,
- $\text{B} = \{-0.2\ m/s^2 \leq \text{"}\textit{lateral acceleration}\text{"} \leq 0.2\ m/s^2\}$,
- $\text{C} = \{\text{"}\textit{lateral acceleration}\text{"} > 0.2\ m/s^2\}$.

This kind of clustering will create a sequence of observation symbols which has been used to estimate the emission matrix in our model.

To estimate the transition probabilities, we have just counted the number of transitions between the three states and normalized each row of the transition matrix to one. To estimate the emission matrix, we have counted the number of times that each observation symbol has been seen in each state.

2.1 Model evaluation

The aim of this study is to find a probabilistic model to recognize the curves. We have estimated the parameters of the model by using a training set and evaluated it by using different new sequences of observations as our test set. To identify the curves for a new lateral acceleration signal, we have considered two different methods as following:

Method 1: Use the estimated transition and emission matrices from the training set.

Method 2: Use the emission matrix from the training set but re-estimate the transition matrix based on the new signal.

The main reason for considering method 2 is the differences between roads that can affect on the transitions between states. The emission matrix describes the property of the curves given certain hidden states, however, the transition matrix describes the duration of the events. Thus, it could be reasonable to update the transition matrix for a new signal to find the hidden states.

2.1.1 Method 1

Here, we have used a training set to estimate the parameters $\lambda = (A, B, \pi)$ of the HMM. The **Viterbi** algorithm, see Viterbi [9] and Forney [4], has been used to find the most probable sequence of hidden states for a new signal.

Suppose that we have classified the new lateral acceleration values with length n and we got an observation sequence $y_1, y_2, ..., y_n$. We would like to find driving events for this new observation. It means that we want to find a sequence of hidden states which maximizes the probability of observing this specified observation. The Viterbi algorithm finds the state sequence $z_1, z_2, ..., z_n$ out of the 3^n possible sequences of length n that maximizes:

$$P(Y_1 = y_1, ..., Y_n = y_n | Z_1 = z_1, ..., Z_n = z_n; \lambda).$$

In fact, the Viterbi algorithm gives a state sequence $z_1, z_2, ..., z_n$ that maximizes the conditional probability of the observation sequence for given parameters $\lambda = (A, B, \pi)$. The result will give the most likely sequence of hidden states from which it is possible to identify the driving events.

2.1.2 Method 2

In this approach, we have fixed the emission matrix from the training set and re-estimated the transition matrix from each new signal. To estimate model parameters based on an observation sequence, we have used the **Baum-Welch** algorithm which was introduced by Baum et al. [1]. It is equivalent to the EM (expectation-maximization) algorithm, see Dempster et al. [3]. The Baum-Welch algorithm is one of the well known methods to estimate the model parameters in HMMs. It is an iterative maximum likelihood method and starts with initial parameters that in our case are set based on training data. The algorithm uses a forward-backward procedure to estimate the model parameters for a given sequence of observations.

Here, we will describe the general EM algorithm following Rabiner [8], and then we will explain what we have used in method 2.

Consider an observation sequence $y_1, y_2, ..., y_T$ with length T and suppose that we have N hidden sates $S = \{S_1, S_2, ..., S_N\}$. We are going to estimate parameters $\lambda = (A, B, \pi)$. The algorithm will start by initial parameters $\lambda_0 = (A_0, B_0, \pi_0)$. It will compute the conditional probability such that:

$$\xi_t(i, j) = P(Z_t = S_i, Z_{t+1} = S_j | Y_1 = y_1, ..., Y_T = y_T; \lambda)$$

which is the probability of being at state i at time t and state j at time $t + 1$ when the observation sequence and the parameters are given.

To compute the probabilities, the forward-backward factors will be used which are defined as the following:

$$\alpha_t(i) = P(Y_1 = y_1, ..., Y_t = y_t, Z_t = S_i; \lambda)$$
$$\beta_t(i) = P(Y_{t+1} = y_{t+1}, ..., Y_T = y_T | Z_t = S_i; \lambda)$$

The formulas for forward-backward recursion are:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(y_{t+1})$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)$$

where $t = 1, 2, ..., T-1$.

To update the model parameters $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, two steps will be considered:

- **E-step**

  In the E-step, we are going to compute $\xi_t(i,j)$. Based on definition of forward-backward factors, $\xi_t(i,j)$ can be written as follows:

$$\begin{aligned}\xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}{P(Y_1 = y_1, ..., Y_T = y_T; \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}\end{aligned}$$

  where $\sum_{t=1}^{T} \xi_t(i,j)$ is the expected number of transitions from state i to j. Let's defne $\gamma_t(i) = P(Z_t = S_i | Y_1 = y_1, ..., Y_T = y_T; \lambda)$, then it would be clear that $\gamma_t(i) = \sum_j \xi_t(i,j)$ and $\sum_{t=1}^{T} \gamma_t(i)$ is exactly the expected number of transitions from state i.

- **M-step**

  In M-step, the parameters $\lambda = (A, B, \pi)$ are going to be updated. Both forward and backward variables will be used to re-estimate model parameters by using the following formulas:

$$\begin{aligned}\bar{\pi}_i &= Expected\ number\ of\ times\ in\ state\ i\ at\ time\ t = 1 \\ &= \gamma_1(i). \\ \bar{a}_{ij} &= \frac{Expected\ number\ of\ transitions\ from\ state\ i\ to\ j}{Expected\ number\ of\ transitions\ from\ state\ i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \\ \bar{b}_j(V_k) &= \frac{Expected\ number\ of\ times\ in\ state\ j\ and\ observing\ V_k}{Expected\ number\ of\ transitions\ in\ state\ j} \\ &= \frac{\sum_{t=1(observing\ V_k)}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.\end{aligned}$$

  By iterating the above procedure, we can improve the probability of a particular observation sequence being generated by the model with given parameters.

In method 2, we didn't update the probabilities $b_j(V_k)$ in the Baum-Welch algorithm since we have fixed the emission matrix $B = B_{training}$. We have just re-estimated the transition matrix and the initial state distribution for a given observation sequence. Then, we have used Viterbi algorithm to find the most likely hidden states based on given parameters.

3 Examples

We have tested our models with simulated and measured data sets. In the first example, we have simulated different lateral acceleration signals as our training and test sets and evaluated the detection of driving events. In the second example, we have used measured data which is dedicated field measurements from a Volvo Truck.

3.1 Simulated lateral acceleration signal

We need a training set to estimate the parameters and a test set to evaluate the model. For this purpose, we have simulated different lateral acceleration signals with sampling period of 1/2 seconds. Figure 2 shows an example of the simulated lateral acceleration signal and the corresponding hidden states. These two simulated signals will be our training set. Next, we will describe how the simulation has been performed.

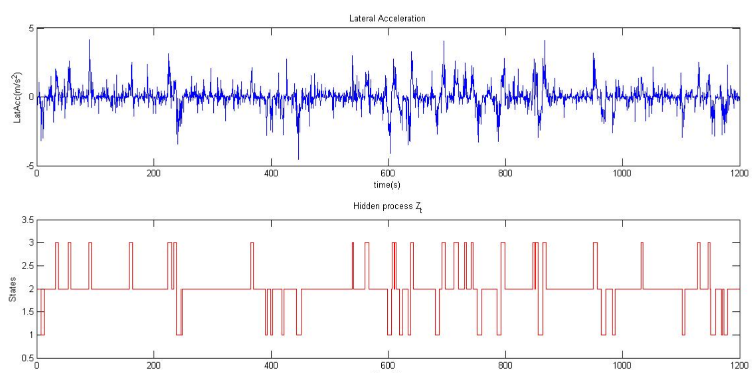

Figure 2: Simulated lateral acceleration signal and the corresponding hidden states.

At first we have generated the events by using a Markov chain in our simulation. We supposed that the probabilities of going from a right turn to a left turn and vice versa are quite small. Most often we will have straight forward after a right turn or a left turn. Thus, we have considered a transition matrix such as

$$\begin{array}{c} \\ \text{R} \\ \text{S} \\ \text{L} \end{array}\begin{array}{c} \begin{array}{ccc} \text{R} & \text{S} & \text{L} \end{array} \\ \begin{pmatrix} 0 & 1-p & p \\ 0.5 & 0 & 0.5 \\ p & 1-p & 0 \end{pmatrix} \end{array}$$

where p=0.1 and represent the probability of going from a right turn to a left turn. We have simulated a Markov chain with three states which represent our sequence of events.

Since we are going to model the length of each **straight** and each **curve**, we have chosen the start and stop points of each $event(i), i = 1, ..., K$ as follows:

$$Start\ point\ for\ event(i) = Stop\ point\ for\ event(i-1),$$
$$Stop\ point\ for\ event(i) = Start\ point\ for\ event(i) + L_i.$$

where $Start\ point\ for\ event(1) = 0$. The length (duration) of each event L_i is random according to specified distributions, namely:

- If $event(i)$ is a curve (right or left turn), then $L_i \sim U(2, 8)$ since each turn may take between 2-8 seconds.
- If $event(i)$ is straight, then $L_i \sim \exp(\theta)$ where $\theta = 20$ shows the average duration of each **straight**.

The result will be our simulated hidden process Z_t.

To generate a lateral acceleration signal, we have used a model suggested by Karlsson [5]. The measured lateral acceleration can be split into two load processes which are the centripetal acceleration and a residual.

Let $a(t)$ be the value of the lateral acceleration at time t, $a_{trap}(t)$ the centripetal acceleration and $a_{res}(t)$ the residual which is a combination of different factors such as driver, road, vehicle and velocity. The model is formulated as:

$$a(t) = a_{trap}(t) + a_{res}(t),$$
$$a_{trap}(t) = v^2(t) * C(t),$$
$$a_{res}(t) = sign(r(t)) * (r(t)^2),$$

where $r(t) \sim Normal(0, 0.5)$ and $C(t)$ is the curvature. In fact for any curve j, the maximum centripetal acceleration is $a_{trap,j} = v_j^2 * C_j$ where v_j indicates the constant speed over curve j and C_j is the maximum curvature. Further, the curvature is modeled by:

$$Y_{C_j} = 1/C_j - r_{turn}$$

where $logY_{C_j} \sim Normal(\mu, \sigma^2)$ and r_{turn} is the turning radius of the vehicle.

By considering suitable values of parameters $\mu = log(1/100)$ and $\sigma = 0.5$ taken from Karlsson [5], we can generate reasonable lateral acceleration signals. To get the Y_t process, we have translated lateral acceleration values into the symbols $V = \{$A, B, C$\}$ as described on page 5.

3.1.1 Estimate of parameters from training set

Recall that in this example the signal in Figure 2 will be our training set. The signal contains 100 events and we have considered the value $\theta = 20$ to get the duration of each straight. Figure 2 illustrates 1200 seconds (2400 samples) of the training set.

To estimate the transition matrix, we have counted the number of transitions between the three states. Finally, we have counted the number of times that each observation symbol (A, B, C) has been seen in each state to estimate the emission matrix.

- The transition matrix:

$$\begin{array}{c c} & \begin{array}{ccc} \text{R} & \text{S} & \text{L} \end{array} \\ \begin{array}{c} \text{R} \\ \text{S} \\ \text{L} \end{array} & \begin{pmatrix} 0.910 & 0.075 & 0.015 \\ 0.010 & 0.979 & 0.011 \\ 0.007 & 0.092 & 0.901 \end{pmatrix} \end{array}$$

- The emission matrix:

$$\begin{array}{c c} & \begin{array}{ccc} \text{A} & \text{B} & \text{C} \end{array} \\ \begin{array}{c} \text{R} \\ \text{S} \\ \text{L} \end{array} & \begin{pmatrix} 0.970 & 0.022 & 0.008 \\ 0.185 & 0.638 & 0.177 \\ 0.014 & 0.034 & 0.952 \end{pmatrix} \end{array}$$

3.1.2 Model evaluation

To recognize the curves for a new simulated lateral acceleration signal, we have considered two different methods. We have generated a new lateral acceleration signal as our test set to compare the two methods. The new signal is shorter than the training set and we have considered the value $\theta = 20$. The simulation contains 28 curves.

Method 1

Here, we have estimated both transition and emission matrices from the training set. Then, the Viterbi algorithm has been used to find the most probable sequence for the new signal. Figure 3 shows the true and detected states based on our model. It can be seen that, for this signal, the method can recognize left turn and right turn without any misclassification error.

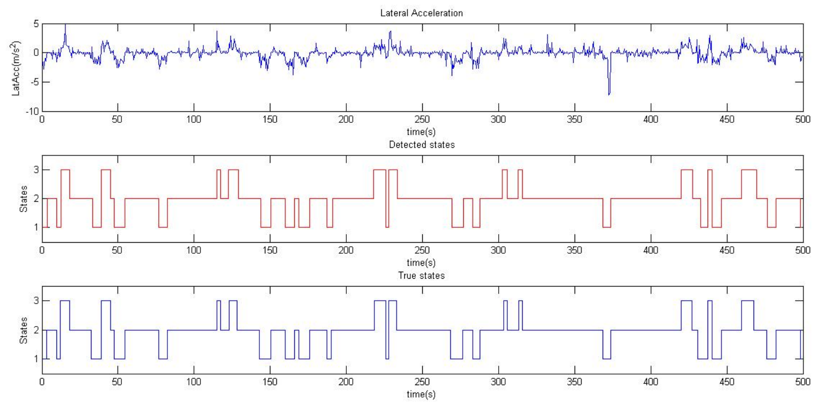

Figure 3: Detection of events using method 1

Method 2

Here, we have used the estimated emission matrix from the training set, but we have estimated the transition matrix from the new signal based on the EM algorithm.

The re-estimated transition matrix:

$$\begin{array}{c c} & \begin{array}{ccc} R & S & L \end{array} \\ \begin{array}{c} R \\ S \\ L \end{array} & \begin{pmatrix} 0.872 & 0.100 & 0.028 \\ 0.029 & 0.957 & 0.014 \\ 0.032 & 0.089 & 0.879 \end{pmatrix} \end{array}$$

The true and detected states for the new signal is shown in Figure 4, where we can see that the misclassification error rate in this case is high.

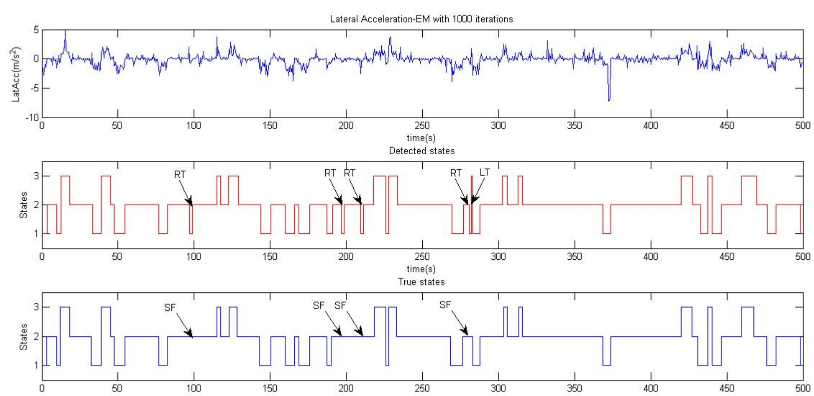

Figure 4: Detection of events using method 2

Comparison between method 1 and 2

To get the misclassification error rates, we have calculated both type I (false positive) and type II (false negative) errors. If we find an event that does not exist, we get a false positive error. However, if we can't detect the true event, the false negative error will happen.

Most of the time, the duration of the detected events are not the same as the real events. Therefore, we have considered the middle time of each detected event and we have compared its label with the true label (hidden state) at that time. The number of times that we got different labels divided by the number of events will be the false positive error rate. Further, to get the false negative error, we have considered the true label of each event at the middle and we have compared it with the detected label.

We have simulated signals with different number of events as our test sets ($n = 10, 100$). We have changed the parameters for the test sets to check how much they will affect on the results. For each case, we performed 1000 simulations to get an average of error rates. Three different duration of straights have been considered by setting $\theta = 4, 20, 100$. For instance, by using $\theta = 100$ we can illustrate a highway. The modified transition probabilities for the Markov chain is:

$$\begin{array}{c c} & \begin{array}{ccc} \mathrm{R} & \mathrm{S} & \mathrm{L} \end{array} \\ \begin{array}{c} \mathrm{R} \\ \mathrm{S} \\ \mathrm{L} \end{array} & \begin{pmatrix} 0 & 1-p & p \\ 0.5 & 0 & 0.5 \\ p & 1-p & 0 \end{pmatrix} \end{array}$$

where the value $p = 0.01, 0.1, 0.25$. If we have larger value of p it means we have more curves in our simulations.

Figure 5 shows the error rates regarding to the different parameters. It can be seen that if we change the p value then the figures do not change very much but if we change the value of θ then there is quite big change between the errors. When we have a small test set, for method 2, the false positive error increases while the false negative decreases. It means that we will detect more events that do not exist.

Generally, the type I error is lower for method 1 but the type II error is higher. If we have a test set of only 10 events, then method 1 should be prefered since we don't have enough data to estimate the parameters.

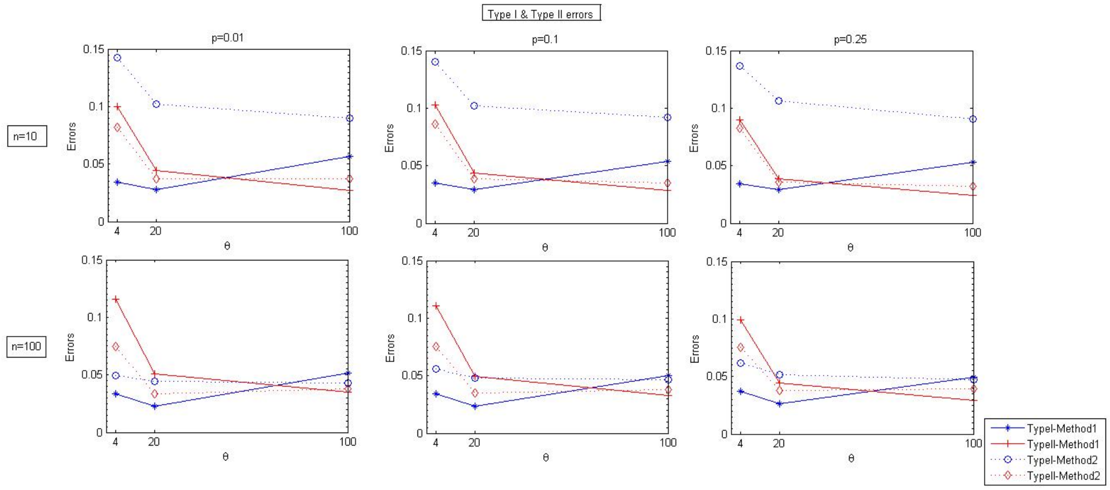

Figure 5: Type I and Type II errors

Based on our simulations we can not draw any clear conclusion but we can have some indications. If the parameters of the test set is similar to the ones in the training set, then method 1 should be preferred. The emission matrix is expected to be similar for all road types. However, the transition matrix should depend on the type of the road. This motivates the use of method 2. For example if we have a lateral acceleration signal from a city road as our training set and we want to detect events based on a lateral acceleration signal from a highway, then the transition matrix from the training set can not be good and it could be re-estimated from the new signal.

The simulation study indicates that method 1 is quite robust to changes in the transition matrix, since it is possible to detect the events even though the transition matrix in training and test sets are different. Method 2 can be used when we have a large enough test set to accurately estimate the transition matrix.

3.2 Comparing HMMs with a simple method

Here, we have used a simple naive algorithm to compare the classification results with HMMs. The aim is to demonstrate the benefit of HMM approach compared to simple threshold based method.

The method detects the curves by using some thresholds for lateral acceleration signal. If the absolute value of lateral acceleration be larger than $0.2m/s^2$ for more than 1.5 seconds, then the algorithm will detect the event as a turning events.

We have just considered method 1 and the simulated lateral acceleration signal has been used for this comparison. Figure 6 shows the results. It can be seen that HMMs works much better than the simple algorithm to recognize the turning events.

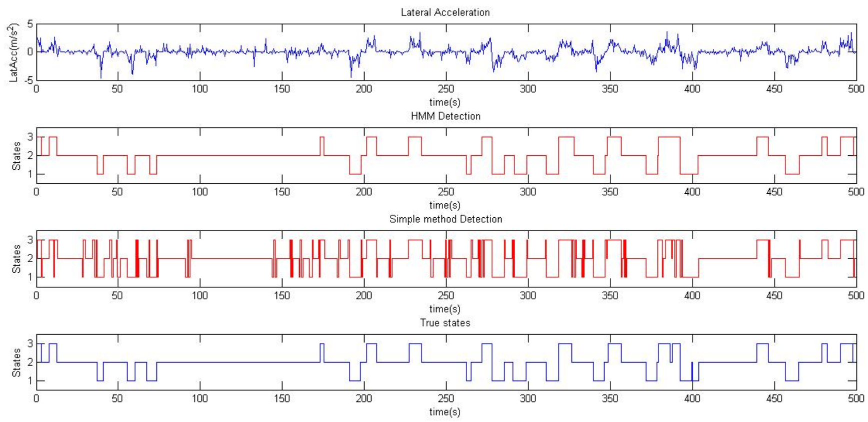

Figure 6: Comparing HMMs with a simple method.

3.3 Measured lateral acceleration signal

The measured data that we have used is a field measurement coming from a Volvo Truck. We have used measured signal from the CAN bus and we have manually detected the events by looking at video recordings from the truck cabin to see what had happened during the driving. By having the start and stop poits of each event, we have created the hidden Z-process.

For the Y-process, we need a lateral acceleration signal which we have computed by using the following formula:

$$lateral\ acceleration = (speed * yaw\ rate)/3.6.$$

To remove the high frequency noise, we have used a Butterworth low-pass filter with 0.5 Hz cut-off frequency. To reduce the amount of data, we have splitted the data into frames (the duration of each frame is 0.5 sec) and calculated the mean value for each frame. We have translated the continuous feature (mean value) into the predefined symbols in each frame by three classes A, B and C where

- A $= \{\text{"}lateral\ acceleration\text{"} < -0.5\ m/s^2\}$,
- B $= \{-0.5\ m/s^2 \leq \text{"}lateral\ acceleration\text{"} \leq 0.5\ m/s^2\}$,
- C $= \{\text{"}lateral\ acceleration\text{"} > 0.5\ m/s^2\}$.

Compared to the clustering in page 6, we have changed the threshold from 0.2 to 0.5 in our clustering in order to improve the detection results.

The signal that is considered has the length 3800 seconds, which we have divided into two parts as our training and test sets. The training set contains 2000 seconds and the test set contains 1800 seconds. Figure 7 shows the training part of the signal and the corresponding manually detected hidden states.

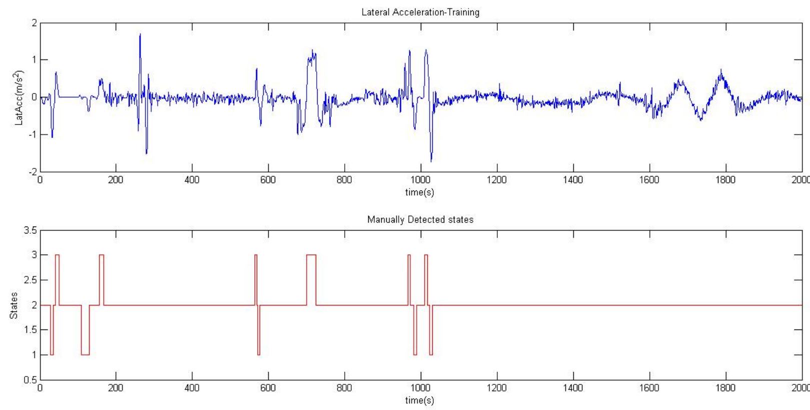

Figure 7: Training part of measured lateral acceleration signal and the corresponding manually detected hidden states.

Method 1

At first, we have used method 1 and estimated transition and emission matrices from the training set, resulting in the transition matrix

$$\begin{array}{cc} & \begin{array}{ccc} R & S & L \end{array} \\ \begin{array}{c} R \\ S \\ L \end{array} & \begin{pmatrix} 0.945 & 0.055 & 0 \\ 0.001 & 0.997 & 0.002 \\ 0 & 0.048 & 0.952 \end{pmatrix} \end{array}$$

and the emission matrix

$$\begin{array}{cc} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} R \\ S \\ L \end{array} & \begin{pmatrix} 0.418 & 0.582 & 0 \\ 0.031 & 0.957 & 0.012 \\ 0 & 0.363 & 0.637 \end{pmatrix} \end{array}$$

The detected states based on method 1 for the test set is shown in Figure 8, where we can compare them with the manually detected states.

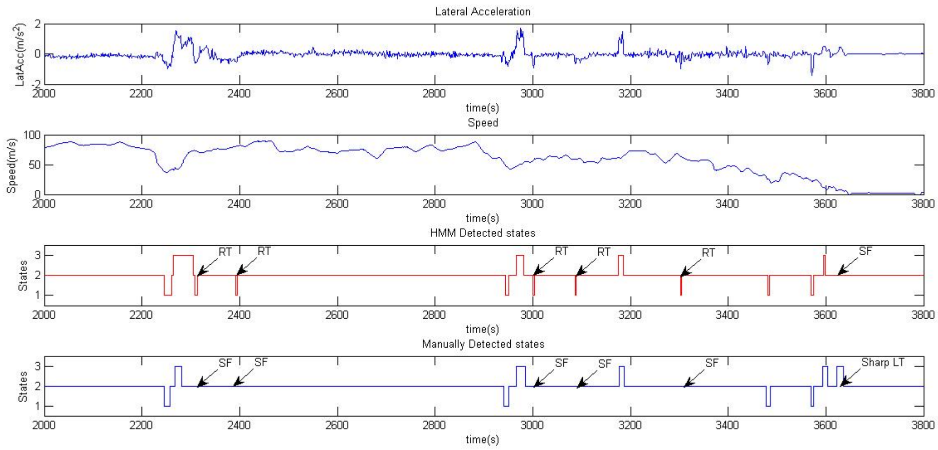

Figure 8: Detection of events in the measured signal using method 1.

It can be seen that the misclassification error rate is high. In all cases, the method can recognize the manually detected curves. However, we have a false positive error since the method has found five right turns that are not in the manual detection. One reason could be that the manual detections are not completely correct because of the visual errors and the low quality of videos. There is also a sharp left turn which could not be detected since the speed is low (about 10 km/s) at that part which makes it hard to recognize the curve correctly.

Method 2

In method 2, the transition matrix have been re-estimated based on the EM algorithm resulting in the estimated transition matrix

$$\begin{array}{cc} & \begin{array}{ccc} \mathrm{R} & \mathrm{S} & \mathrm{L} \end{array} \\ \begin{array}{c} \mathrm{R} \\ \mathrm{S} \\ \mathrm{L} \end{array} & \begin{pmatrix} 0.895 & 0.105 & 0 \\ 0.004 & 0.995 & 0.001 \\ 0 & 0.033 & 0.967 \end{pmatrix} \end{array}$$

Figure 9 shows the results based on method 2. It can be seen that the results are the same as for method 1. Since the road type of the test set is similar to the one in the training set, method 1 and 2 performed similarly to detect the events.

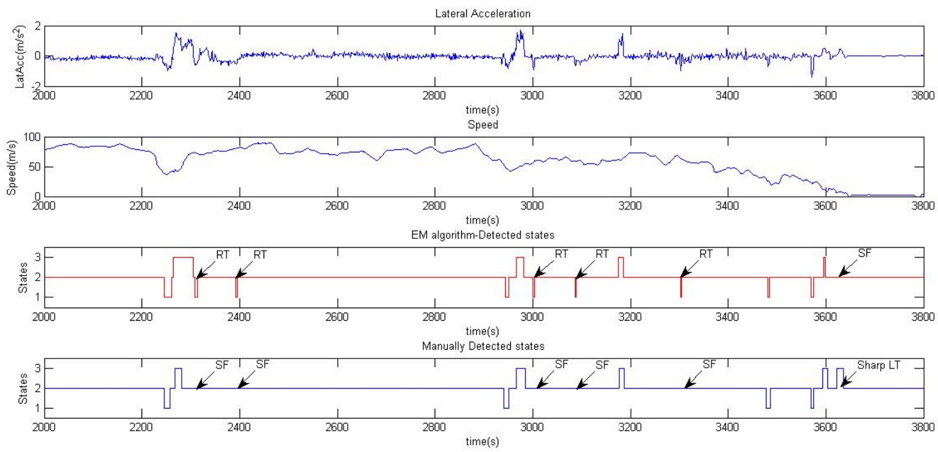

Figure 9: Detection of events in the measured signal using method 2.

4 Conclusion

The examples in this study indicate that the HMMs can be used to recognize the curves based on a lateral acceleration signal. We have considered three driving events (right turn, left turn and straight forward) as the hidden states and constructed the model based on them. The parameters of the model have been estimated by considering two different methods. In method 1, we have estimated the transition and emission matrices from the training set, while in method 2 the emission matrix has been fixed from the training set and the transition matrix has been re-estimated based on the test set. The results of the simulation study show that method 1 should be preferred if the parameters of the test set is similar to the ones in the training set, i.e. the characteristics of the roads are likely to be similar. The emission matrix is expected to be similar for all road types, however, the transition matrix should depend on the type of the road. This motivates the use of method 2. If we have a lateral acceleration signal from a city road as our training set and we want to detect events based on a lateral acceleration signal from a highway, then the transition matrix from the training set can not be good and it could be re-estimated from the new signal.

The method can be extended to detect more events, such as braking and static steering, by considering more signals which contain useful information about the events. One approach can be to combine the essential signals and increase the number of classes in Y-process. It could then be possible to detect more events.

Acknowledgment

We are thankful to Dr. Thomas Svensson for his useful ideas and helpful suggestions in this study. We would like to thank Volvo Trucks for supplying the data in this study and to the members in our research group at Volvo for their valuable advice. Finally, we gratefully acknowledge the financial support from VINNOVA.

References

[1] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, Ann. Math. Statist., vol. 41, no. 1, pp. 164-171, 1970.

[2] H. Berndt, K. Dietmayer *Driver Intention Inference with Vehicle Onboard Sensors*, IEEE International Conference on Vehicular Electronics and Safety (ICVES), Nov. 11-12, Pune, 2009, 102-107.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin *Maximum likelihood from incomplete data via EM algorithm*, J. Roy. Stat. Soc., vol.39, no. 1, pp. 1-38, 1977.

[4] G. D. Forney, *The Viterbi algorithm*, Proc. IEEE vol.61, pp. 268-278, Mar. 1973.

[5] M. Karlsson,*Load Modelling for Fatigue Assessment of Vehicles-a Statistical Approach*, PhD Thesis, Chalmers University of Technology, 2007

[6] D. Mitrović, *Learning Driving Patterns to Support Navigation* , PhD Thesis, University of Canterbury, 2004

[7] D. Mitrović, *Reliable Method for Driving Events Recognition*, IEEE Trans. Syst., vol. 6, no. 2, June 2005, 198-205.

[8] Lawrence R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, IEEE, VOL. 77, NO. 2, Page(s) 257-286, February 1989

[9] A. J. Viterbi, *Error bounds for convolutional codes and an asymptotically optimal decoding algorithm*, IEEE Trans. Informat. Theory, vol. IT-13, pp. 260-269, Apr. 1967.