THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Hidden Markov models for detecting steering events and evaluating fatigue damage

Roza Maghsood

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Division of Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2016

Hidden Markov models for detecting steering events and evaluating fatigue damage
Roza Maghsood
ISBN 978-91-7597-414-9

Doktorshavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4095
ISSN 0346-718X

Division of Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31 772 1000

Typeset with LaTeX.
Printed in Göteborg, Sweden 2016

Hidden Markov models for detecting steering events and evaluating fatigue damage

Roza Maghsood

Division of Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

For fatigue design of vehicles, the external loads need to be assessed. One source of variation in the loads is the driver's behaviour. This behaviour can be characterized as driving events. In this study we use hidden Markov models (HMMs) for identifying the driving events. The idea is that one can see the driving events as hidden states and construct the Markov model based on them. The steering events such as curves and maneuvers are detected using on-board logging signals available on trucks. We test our proposed method for both discrete and continuous observation processes. The EM algorithm has been used for estimating the HMM parameters. Also a recursive EM algorithm has been proposed for on-line estimation. A fatigue damage index is computed by using the extreme forces occurring during those steering events. An explicit formula for the expected damage is found using vehicle independent driving events for steering components.

Keywords: Hidden Markov models (HMMs), online EM algorithm, fatigue damage, vehicle independent load models, steering events, on-board logging signals, lateral loads.

Acknowledgements

First, I would like to express my special appreciation and thanks to my academic supervisors, Igor Rychlik, Pär Johannesson and Jonas Wallin, for their generous help and support of my Ph.D study and research. Your advice on both my research as well as on my career have been priceless.

I would like to express my sincere gratitude to my industrial supervisor Bengt Johannesson, for his constant support and encouragement to increase my knowledge about automotive applications. Special thanks go to my industrial co-supervisor, Gwenaëlle Le Corre, for all her help and support during this work.

I would also like to thank Volvo Trucks for supplying the data in this study and give a special gratitude to the members in our research group, Jacques de Maré, Magnus Karlsson, Hans Fredrik Henrysson, Stefan Edlund, Fredrik Öijer, Thomas Svensson, Martin Olofsson, Tatiana Smirnova and Andreas Emilsson for their great ideas and their valuable comments.

Furthermore, I gratefully acknowledge the financial support from VINNOVA.

Many thanks go to my excellent teachers at the department of Mathematical Sciences, specially Christer Borell, Rebecka Jörnsten and Holger Rootzen. I learned alot from my scientific interaction with you. Thanks for being such a nice person and a great teacher!

I would further like to thank Petter Mostad, Serik Sagitov and Aila Särkkä for all their guidance and consults during my PhD studies. Special thanks go to Lotta Fernström, Marie Kühn, Cecilia Gelin, and Jovan Pankovski, for all their help with the administrative stuff.

Thanks to all my wonderful friends and colleagues at the department, particularly Malin, Sandra, Mariana, Henrike, Fredrik, Viktor, Tobias and many more, for their support and their friendliness. Big thanks to my dear friends outside the department, specially Shirin, Pegah, Reza, Mahdi, Maryam and Mahdis. You were always beside me during the happy and hard moments to push and motivate me.

Finally, I acknowledge the people who mean a lot to me, my parents, Paeizeh and Hossein, for all their love, support and encouragement and my brother, Ramin, for always supporting me and being my best friend. Words cannot express how grateful I am to having you. I would also like to thank my uncle Hassan, my lovely cousins Peyman, Parisa and her husband Stefan, for supporting and making good moments throughout my life in Sweden.

Roza Maghsood
Göteborg, May 20, 2016

To

my beloved parents

and

my lovely brother

List of papers

The thesis includes the following papers:

I. **R. Maghsood**, P. Johannesson, *Detection of Steering Events based on Vehicle Logging Data using Hidden Markov Models*, International Journal of Vehicle Design,Vol 70, Page: 278 – 295, 2016.

II. **R. Maghsood**, I. Rychlik, J. Wallin, *Modeling Extreme Loads Acting on Steering Components using Driving Events*, Probabilistic Engineering Mechanics,Vol 41, Page: 13 – 20, 2015.

III. **R. Maghsood**, I. Rychlik, P. Johannesson, *Load Description and Damage Evaluation using Vehicle Independent Driving Events*, Procedia Engineering, Vol 101, Page: 268 – 276, 2015. [Conference paper - peer reviewed]

IV. **R. Maghsood**, P. Johannesson, J. Wallin, *Detection of Steering Events using Hidden Markov Models with Multivariate Observations* (submitted for publication), 2016

V. **R. Maghsood**, J. Wallin, *On-line Estimation of Driving Events and Fatigue Damage on Vehicles* (submitted for publication), 2016

My contribution to the appended papers:

I. I wrote the paper under the close supervision of Dr. Pär Johannesson.

II. I wrote the paper and developed the algorithms under the supervision of Prof. Igor Rychlik and Dr. Jonas Wallin.

III. I wrote the paper under the supervision of Prof. Igor Rychlik and Dr. Pär Johannesson.

IV. I wrote the paper under the supervision of Dr. Pär Johannesson and Dr. Jonas Wallin.

V. I wrote the paper and developed the on-line algorithm under the close supervision of Dr. Jonas Wallin.

Contents

1 Introduction

From an industrial point of view, reliability means that the products comply to customer demands. Companies want to design a vehicle based on customer usage. By enabling the optimization of vehicles with regard to the use, the companies obtain benefits such as increased capacity, reduced fuel consumption, reduced material consumption in production and reduced road wear. All these points can be summarized as reducing environmental impact. Knowing the customer driving habits – like the average number of left and right turns – is very important in vehicle design. These behaviors together with the expected damage caused by an event can be used to design a robust component.

A driver can affect the load by changing the speed, braking or adapting to the curves. These behaviors are characterized as driving events. In general, the customer behavior is unknown and needs to be estimated. To identify the events we need to use the information available for all vehicles by means of CAN (Controller Area Network) bus data. Since the driving events are not recorded in CAN data, we develop a method for identifying the driving events, their frequency for different uses of the vehicle. In addition, finding the distribution of severity within each event is of interest. For example, for curves the damage can be calculated by finding the distribution of maximum forces for each event.

There are different methods for modeling driver actions and identifying driving events. For instance, Nilsson et al. (2014) have developed an on-line cycle detection algorithm to identify driving events during vehicle operation, by using only production sensors. Karlsson (2007) has described customer usage by classifying the type of roads. This is important since different aspects of the customer usage are relevant in order to determine the fatigue damage in service. Such aspects are the type of roads, transport mission, driver's behavior and different kind of maneuvers they perform.

In this thesis, a method is presented to detect the driving events using Hidden Markov Models (HMMs), see e.g. Rabiner (1989) and Cappé et al. (2005) for a general description of HMMs. The HMMs are the reliable and robust methods for event recognition. The idea of using HMMs to identify driving events is not new and it has been used in many applications, see for example Mitrović (2004, 2005) and Berndt and Dietmayer (2009). They constructed one HMM for each type of driving event such as left and right curves, left, right and straight on roundabout. In our study, we have used a single HMM for describing all driving events.

We detect the steering events using an HMM, where the hidden process is the driving states and the observations are derived from CAN signals. We test our proposed method for both discrete and continuous observation processes with univariate and multivariate distributions. In the continuous version, the generalized Laplace model (GAL) is proposed as the distribution of the observation sequence in the HMM. Laplace models have previously been used to describe road roughness and responses measured on driving vehicles, see Kvanström et al. (2013) and Bogsjö et al. (2012).

We use two types of parameter estimation methods for HMMs, which are off-line and on-line EM algorithms. The on-line algorithm gives us the opportunity to track the driver's actions over time. We use an on-line EM-algorithm for parameter estimation, which is a modification of the usual EM algorithm. In order to achieve this goal, a recursive implementation of the EM algorithm will be used to construct a method which never stops the adaptation.

In our study, we are interested in the number of occurrences of steering events for a costumer. Together with the forces generated by the events, one can calculate the expected damage of a component for a specific vehicle. We investigate the damage caused by driving events and we compute the expected damage using the on-line estimation of the transition matrix.

Vehicle independent load description is presented in Section 2. In Section 3, we describe the concept of HMMs. Damage definition and the proposed model to calculate the expected damage are described in Section 4. Summary of papers are given in Section 5. Conclusions are presented in Section 6.

2 Vehicle independent driving events

For vehicle companies, it is important to characterize the usage of the trucks independent of the vehicle. To have components that are strong enough, they need to describe the load environments. The loads will be different for different usage of trucks and for different driver's behavior. Measuring the load on each truck is expensive. However, the companies want to measure and identify activities of the driver and specify the relevant events occurring on the road. By counting the number of driving events, they can estimate the fatigue damage caused by the same kind of events.

Due to the lack of storage capability, gathering and storing all information about the customers and measurement signals is not possible. Finding the vehicle independent part of the load, e.g. frequencies of driving events, is an important part of the description of vehicle independent service loadings. In this study we are interested in vehicle independent driving events such as curves and maneuvers which are relevant for steering components. These kinds of events are called steering events and generate high forces in the steering arms.

2.1 Vehicle logging data

In this study a method will be proposed for detection of steering events such as curves and maneuvering using on-board logging signals available on trucks. This information is available for all vehicles and can be obtained from CAN (Controller Area Network) bus data. Since the driving events are not recorded in CAN-data, one needs to estimate them. If we define the events by using the information from CAN-data, we can detect the number of events that occur in customer vehicles.

There are more than 80 signals from CAN bus data that can be used to identify the driving events. The signals we have used from CAN-data are:

- Steering wheel angles,
- Vehicle speed,
- Yaw rate.

The steering wheel is the wheel we hold in our hands while we drive and the angle is defined as the angle deviation from driving straight ahead.

The yaw rate contains information about steering events only for a non-zero speed. By steering, the entire vehicle will shift direction and this will happen at a certain angular velocity, which is the yaw rate.

We have used the yaw rate and speed to get an accurate lateral acceleration signal which is computed by the following formula:

$$\text{"}\textit{lateral acceleration}\text{"} = \text{"}\textit{speed}\text{"} \cdot \text{"}\textit{yaw rate}\text{"}.$$

The lateral force is proportional to the lateral acceleration in turning events. For most components these loads are not as damaging as the vertical loads, but they have a large impact on steering components Karlsson (2007).

2.2 Steering events

In this study we are interested in steering events that are relevant for steering components. We divide the steering events into curves and maneuvers:

- The curve events give rise to lateral forces through lateral acceleration and occur when the vehicle drives at a speed higher than about 10 km/h. We limit the analysis to three cases: right turn, left turn and straight forward for turns.
- The maneuvering events generate high forces due to steering at low speed, typically lower than 10 km/h, e.g. driving in or out of a parking lot, standing still but turning steering wheel. Three maneuvering events are considered: steering right, steering left and straight forward for maneuvers.

3 Hidden Markov models

Hidden Markov models are statistical models often used in signal processing, such as speech recognition and modeling the financial time series, see for instance Cappé et al. (2005) and Frühwirth-Schnatter (2006). An HMM is a bivariate Markov process $\{Z_t, Y_t\}_{t=0}^{\infty}$ where the underlying process Z_t is an unobservable Markov chain and is observed only through the Y_t. The observation sequence Y_t given Z_t is a sequence of independent random variables and the conditional distribution of Y_t depends only on Z_t.

The main parameters in an HMM are the transition matrix, the emission distribution, and the initial state distribution. All parameters must be estimated to characterize the model.

Definition 1. (Transition matrix) Let Z_t takes values on a discrete space $\{1, 2, \ldots, N\}$. The transition probabilities between the hidden states are defined by transition matrix $\boldsymbol{Q} = (q(i, j))$, where:

$$q(i, j) = P(Z_{t+1} = j | Z_t = i), \; i, j = 1, 2, ..., N. \tag{1}$$

is the probability of transition from state i to state j and $\sum_{j=1}^{N} q(i, j) = 1$.

Definition 2. (Emission distribution) The emission distribution is the conditional distribution of observations Y_t given states Z_t. The observed variable Y_t can be a discrete, continuous, univariate or multivariate variable. For a discrete Y_t, the emission distribution is a probability distribution of observation symbols, while for a continuous Y_t it can be described by the parameters of the conditional distribution of Y_t given Z_t.

Definition 3. (Initial state probabilities) In an HMM, the state where the hidden process will start is modeled by the initial state probabilities $\boldsymbol{\pi} = (\pi_i)$, where π_i is:

$$\pi_i = P(Z_0 = i), \; i = 1, 2, ..., N \tag{2}$$

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

3.1 HMMs for steering events

The principle aim of this study is to estimate the number of steering events, such as curves and maneuvers, occurring on the road. We propose using HMMs to detect steering events by using on-board logging signals available on trucks, such as lateral acceleration, vehicle speed and steering wheel angle.

Suppose that there are three driving states right turn (1 = "RT"), straight forward (2 = "SF") and left turn (3 = "LT"). The idea is to construct an HMM based on these three states and to identify the activities of the driver and specify the relevant events. Figures 1 and 2 illustrate three hidden states {RT, SF, LT} and the transitions between them with the emission distribution of the observations.

3.1.1 Discrete model

In a discrete model, the observation process is discretized by using thresholds. Let the sequence of observation have the possible values $V = \{V_1, V_2, ..., V_M\}$. The probability distribution of observation symbols in each state is usually presented by the emission matrix, $B = \{b_j(V_k)\}$, where

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

and $\sum_{k=1}^{M} b_j(V_k) = 1$. Figure 1 illustrates an HMM with three hidden states {RT, SF, LT} which can emit three discrete symbols V = {A, B, C}. For instance, the lateral acceleration signal which is used to detect the curves, can be translated into predefined classes V = {A, B, C}, as follows:

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

where the threshold 0.2 m/s^2 has been chosen based on experience. This kind of clustering will create a sequence of observation symbols which will be the observation sequence $\{Y_t\}_{t=0}^{\infty}$.

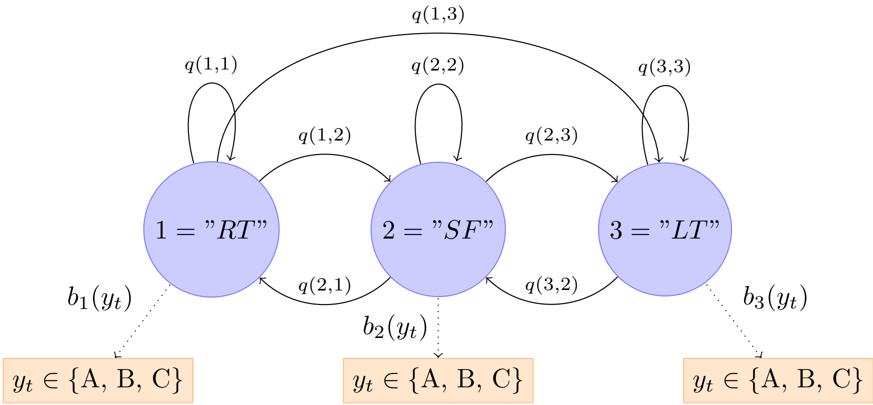


Figure 1: An HMM with three hidden states {RT, SF, LT} which can emit three discrete symbols V={A, B, C} at time t. The parameter $q(i, j)$ is the conditional probability of transition from state i to state j, and $b_j(y_t)$ is the emission probability of symbol y_t given state j.

3.1.2 Continuous model

In a continuous version of HMMs, the actual observation sequence $\{Y_t\}_{t=0}^{\infty}$ will be used. The continuous model has two main advantages over the discrete version. Firstly, in the discrete model one has to manually set the threshold levels, whereas the continuous model is entirely estimated from the actual data. Secondly, the continuous model can easily be extended to incorporate multivariate sources of information, which is not easy in the discrete threshold approach.

The conditional distribution of Y_t given Z_t is denoted by:

$$g_{\boldsymbol{\theta}}(i, y_t) = f_{Y_t}(y_t | Z_t = i; \boldsymbol{\theta}), \; i = 1, 2, ..., N, \; y_t \in \mathbb{R}, \tag{3}$$

where $\boldsymbol{\theta}$ represents the parameter vector of distribution, and y_t is the observed value of Y_t. Figure 2 illustrates the construction of an HMM with three hidden states {RT, SF, LT} where the actual lateral acceleration signal has been used as observation Y_t. In our case, the Laplace distribution is used to model the observations in each state. The generalized asymmetric Laplace distribution (GAL) is defined in Section 3.2.

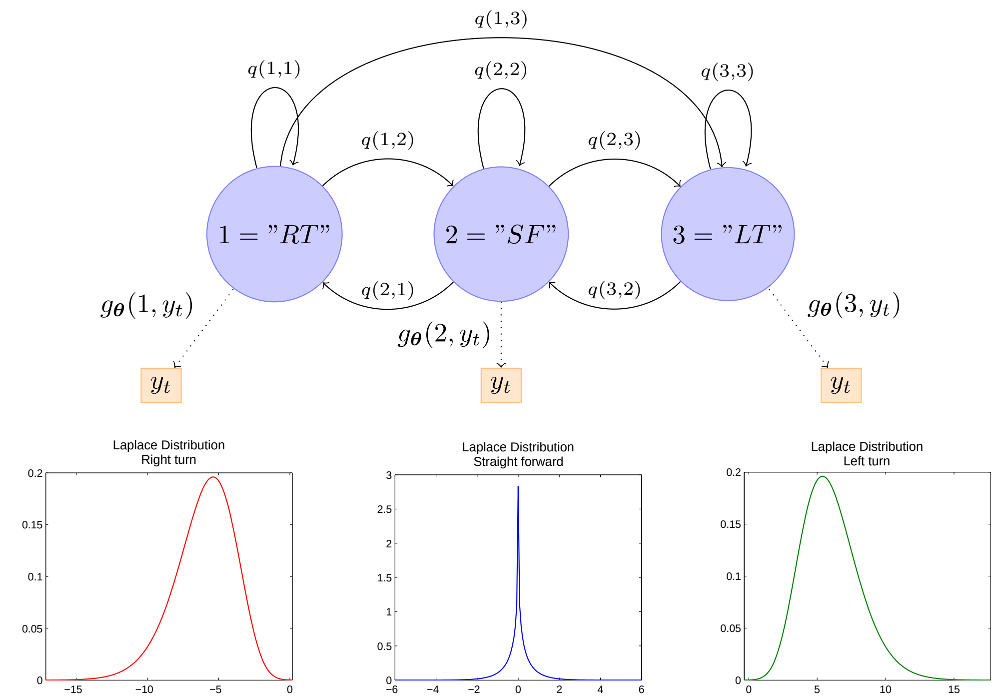


Figure 2: An HMM with three hidden states {RT, SF, LT} which the observation has Laplace distribution in each state. The parameter $q(i, j)$ is the conditional probability of transition from state i to state j, and $g_{\boldsymbol{\theta}}(i, y_t)$ is the conditional distribution of Y_t given state i.

3.1.3 A combination of continuous and discrete distributions

As mentioned above, the observations can be multivariate random variables. The aim is to estimate the model parameters which maximize the likelihood function. In order to do this, we need to find the conditional distribution of $\boldsymbol{Y}_t$ given Z_t. Suppose that the observation $\boldsymbol{Y}_t = (Y_{t,1}, ..., Y_{t,d})$ is a multivariate time series with $d = d_1 + d_2$ dimensions. The first d_1 random variables $(Y_{t,1}, ..., Y_{t,d_1})$ are continuous with their observed values $(y_{t,1}, ..., y_{t,d_1})$, and have the joint probability density function $f_{Y_{t,1},...,Y_{t,d_1}}(y_{t,1}, ..., y_{t,d_1}|Z_t = i; \boldsymbol{\theta}_1)$. We assume that the last d_2 observations $(Y_{t,d_1+1}, ..., Y_{t,d_2})$ are discrete variables and have the joint probability mass function $P(Y_{t,d_1+1} = y_{t,d_1+1}, ..., Y_{t,d_2} = y_{t,d_2}|Z_t = i; \boldsymbol{\theta}_2)$. For the sake of simplicity we assume that the continuous and discrete variables are conditionally independent given the hidden state. Then, the likelihood function of $\boldsymbol{Y}_t$ given Z_t for a set of parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ is as follows:

$$\begin{aligned} g_{\boldsymbol{\theta}}(i, \boldsymbol{y}_t) &= f_{Y_{t,1},...,Y_{t,d_1}}(y_{t,1},...,y_{t,d_1}|Z_t = i; \boldsymbol{\theta}_1) \\ &\qquad P(Y_{t,d_1+1} = y_{t,d_1+1},...,Y_{t,d_2} = y_{t,d_2}|Z_t = i; \boldsymbol{\theta}_2), \quad (4) \end{aligned}$$

where $\boldsymbol{y}_t = (y_{t,1},...,y_{t,d})$ is the observed value of $\boldsymbol{Y}_t$ and $i = 1, 2, ..., N$.

Note that any continuous or discrete distributions can be used in Eq. (4). In our study, different on-board logging signals can be used as observations $\boldsymbol{Y}_t$ and some of them are discretized into the predefined levels. In papers I, II and III, we separate the curve and maneuvering events and construct two different models, HMM_C for the curves and HMM_M for the maneuvers. Another approach can be to combine the two models into a larger HMM containing both curve and maneuvering events. In paper IV, we investigate the usefulness of HMMs with multidimensional observations, for detection of all simultaneously steering events.

To identify all steering events, we use a single HMM with a combination of continuous and discrete distributions for observations. We consider six states as follows: right turn (1 = "RT"), straight forward for turns (2 = "SFT") and left turn (3 = "LT"), steering right (4 = "SR"), straight forward for maneuvers (5 = "SFM") and steering left (6 = "SL"). We use three signals: lateral acceleration ($Y_{t,1}$), steering angle speed ($Y_{t,2}$) and vehicle speed ($Y_{t,3}$). We assume that $Y_{t,1}$ and $Y_{t,2}$ are independent random variables with Laplace distribution and we discretize $Y_{t,3}$ into the three levels:

- $1 = \{0$ km/h $\leq$ "*speed*" < 1 km/h$\}$,
- $2 = \{1$ km/h $\leq$ "*speed*" < 10 km/h$\}$,
- $3 = \{$"*speed*" ≥ 10 km/h$\}$.

We use a combination of Laplace and discrete distributions for observations and construct the HMM. Figure 3 shows an example of the measured lateral acceleration, steering angle speed and speed signals with the corresponding detected hidden states process. The signals are dedicated field measurements from a Volvo Truck. The sequence of hidden states are reconstructed using the Viterbi algorithm proposed by Viterbi (1967).

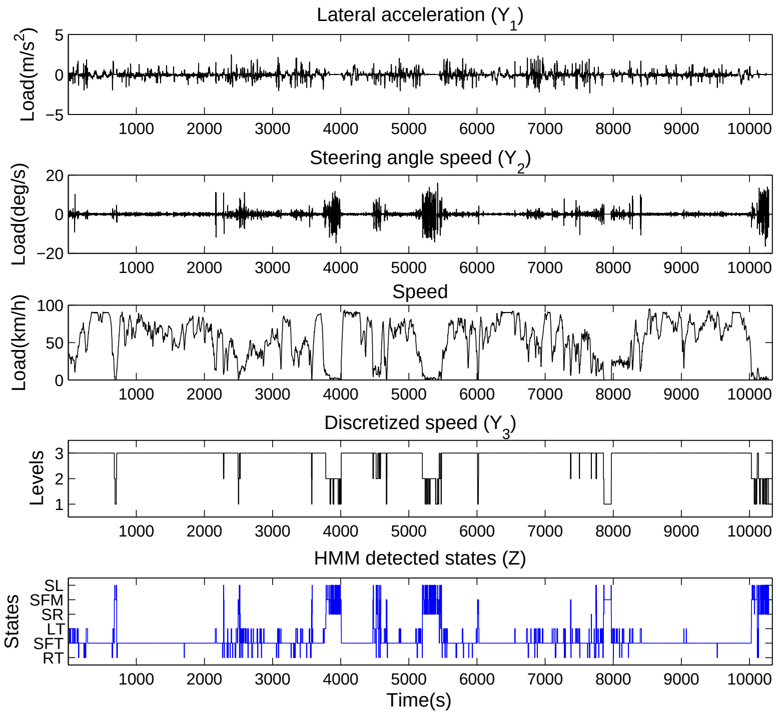


Figure 3: Lateral acceleration, steering angle speed and speed signals with the corresponding steering states detected by HMM.

3.2 The generalized asymmetric Laplace distribution

The generalized asymmetric Laplace distribution (GAL), see Kotz et al. (2001), is a flexible distribution with four parameters: $\boldsymbol{\delta}$, location vector, $\boldsymbol{\mu}$, shift vector, $\nu > 0$, shape parameter, and $\boldsymbol{\Sigma}$, scaling matrix, and denoted by $GAL(\boldsymbol{\delta}, \boldsymbol{\mu}, \nu, \boldsymbol{\Sigma})$. The probability density function (pdf) of a $GAL(\boldsymbol{\delta}, \boldsymbol{\mu}, \nu, \boldsymbol{\Sigma})$ distribution is:

$$\begin{aligned} g(\boldsymbol{y}) \quad &= \quad \frac{1}{\Gamma(1/\nu)\sqrt{2\pi}} \left(\frac{\sqrt{(\boldsymbol{y}-\boldsymbol{\delta})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}-\boldsymbol{\delta})}}{c_2} \right)^{\frac{1/\nu - d/2}{2}} e^{(\boldsymbol{y}-\boldsymbol{\delta})\Sigma^{-1}\boldsymbol{\mu}} \\ & \qquad K_{1/\nu - d/2}\left(c_2 \sqrt{(\boldsymbol{y}-\boldsymbol{\delta})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}-\boldsymbol{\delta})} \right), \end{aligned}$$

where d is the dimension of $\boldsymbol{Y}$, $c_2 = \sqrt{2 + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}$ and $K_{1/\nu - d/2}(.)$ is the modified Bessel function of the second kind. The normal mean variance mixture representation can give an intuitive feel of the distribution. This is a random variable $\boldsymbol{Y}$ having GAL distribution, and the following equality holds:

$$\boldsymbol{Y} \stackrel{d}{=} \boldsymbol{\delta} + \Gamma\boldsymbol{\mu} + \sqrt{\Gamma}\boldsymbol{\Sigma}^{1/2}\boldsymbol{Z},$$

where Γ is a Gamma distributed random variable with shape $1/\nu$ and scale one, and $\boldsymbol{Z}$ is a vector of d independent standard normal random variable. For more details see Barndorff-Nielsen et al. (1982).

In this study, Laplace distribution has been fitted to lateral acceleration (Y_1) and steering angle speed (Y_2). There are two choices for modeling the observations $\boldsymbol{Y} = (Y_1, Y_2)$ with Laplace distribution. We can either consider that Y_1 and Y_2 are dependent, which means that they have the same shape parameter ν, or we can assume that they are independent but with different ν. Both cases have been tested. It has been found that having separate ν for Y_1 and Y_2 gives better fit to data and also better estimation of the transition matrix.

3.3 Estimation of HMM parameters

As was mentioned previously, $\Theta = (\boldsymbol{Q}, \boldsymbol{\theta})$ denotes the set of HMM parameters, where $\boldsymbol{Q} = (q(i,j))$ is the transition matrix of Markov chain Z_t, and $\boldsymbol{\theta}$ denotes the parameters of the conditional distribution of Y_t given Z_t. We mainly focus on estimating the transition matrix $\boldsymbol{Q}$ of HMM based on the observation sequences. The parameter $\boldsymbol{\theta}$ is estimated through a maximum likelihood estimation on a training set. This is because the conditional distribution of Y_t given Z_t in our case study represents the vehicle specific data which can be estimated under well-defined conditions on the proving ground. However, the differences between types of roads can affect the transition probabilities and a transition matrix describes the duration of the events. Therefore, it could be relevant to update the transition matrix for a new signal to find the hidden states.

We use the EM algorithm for parameters estimation. The EM algorithm is a common method for estimating the parameters in HMMs. It is an optimization algorithm used to find the parameters that maximize the likelihood. The algorithm is both robust and often easy to implement.

We apply two types of EM algorithms: off-line and on-line estimations. In the off-line algorithm, all observations Y_t will be used to estimate the transition matrix, while in the on-line version the parameters are updated recursively each time a new observation is made. A forgetting factor will be used in the on-line algorithm which represents the influence of the past data. Figure 4 illustrates both off-line and on-line procedures to estimate the transition matrix.

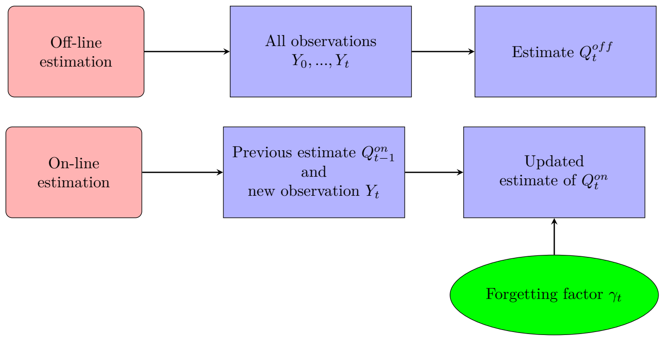


Figure 4: Off-line vs on-line estimation.

3.3.1 EM algorithm

We want to estimate the model parameters which maximize the likelihood function. The expectation-maximization (EM) algorithm introduced by Dempster et al. (1977) is an iterative algorithm that maximizes the likelihood starting from an initial guess. The EM algorithm can be used to estimate the HMM parameters. In an HMM, we have two processes Z_t and Y_t, where Z_t is not possible to measure. Here, the sequence of hidden states Z_t takes values on a discrete space $\{1, 2, \ldots, N\}$. We assume that $y_0, ..., y_T$ are observed, and $z_0, ..., z_T$ are referred as hidden data. Therefore, the complete data likelihood is given by:

$$p(y_0, ..., y_T, z_0, ..., z_T; \Theta) = \pi_{z_0} g_{\boldsymbol{\theta}}(z_0, y_0) q(z_0, z_1) g_{\boldsymbol{\theta}}(z_1, y_1) ... q(z_{T-1}, z_T) g_{\boldsymbol{\theta}}(z_T, y_T). \tag{5}$$

However, the likelihood available for estimating parameters is the joint probability density function of $Y_0, ..., Y_T$, calculated by:

$$L(\Theta) = p(y_0, ..., y_T; \Theta) = \sum_{z_0=1}^{N} ... \sum_{z_T=1}^{N} p(y_0, ..., y_T, z_0, ..., z_T; \Theta). \tag{6}$$

Computing the sums in the likelihood function $L(\Theta)$ is not numerically feasible. Thus, direct maximization of the likelihood is not computationally tractable. The EM algorithm provides a method for estimating the HMM parameters by using the expected value of the complete data log-likelihood given known observations and current parameters. The algorithm starts with an initial guess of the parameters $\Theta^{(0)}$, and then iteratively updates the current parameters by maximizing:

$$\begin{aligned}
\tau(\Theta, \Theta^{(n)}) &= E\left[\log p(Y_0, ..., Y_T, Z_0, ..., Z_T; \Theta) \big| y_0, ..., y_T; \Theta^{(n)}\right] \\
&= E\left[\log \pi_{z_0} \big| y_0, ..., y_T; \Theta^{(n)}\right] \\
&+ \sum_{l=0}^{T-1} E\left[\log q(z_l, z_{l+1}) \big| y_0, ..., y_T; \Theta^{(n)}\right] \\
&+ \sum_{l=0}^{T} E\left[\log g_{\boldsymbol{\theta}}(z_l, y_l) \big| y_0, ..., y_T; \Theta^{(n)}\right],
\end{aligned}$$

for $n = 0, 1, 2, ...$ until convergence. Thus, the n^{th} iteration of the EM algorithm consists of the following two steps:

- The E-step, where the expected complete log-likelihood $\tau\left(\Theta, \Theta^{(n)}\right)$ is computed,
- The M-step, where the maximum likelihood estimate of the parameter $\Theta^{(n+1)} = \text{argmax}_{\Theta}\, \tau\left(\Theta, \Theta^{(n)}\right)$ is computed.

For our specific model, the parameter of interest is $\boldsymbol{Q} = (q(i, j))$. In this case, the E-step consists of computing $P(Z_{t-1} = i, Z_t = j | y_0, ..., y_T; \boldsymbol{Q}^{(n)})$, which is the conditional probability of being at state i at time $t-1$ and state j at time t when the observation sequence and the parameters are given. Further, the M-step uses the conditional probability to compute the transition probabilities as follows:

$$\begin{aligned}
q^{(n+1)}(i, j) &= \frac{\text{"}Expected\ number\ of\ transitions\ from\ state\ i\ to\ j\text{"}}{\text{"}Expected\ number\ of\ visits\ to\ state\ i\text{"}} \\
&= \frac{\sum_{l=0}^{T-1} P(Z_l = i, Z_{l+1} = j | y_0, ..., y_t; \boldsymbol{Q}^{(n)})}{\sum_{l=0}^{T} P(Z_l = i | y_0, ..., y_t; \boldsymbol{Q}^{(n)})}.
\end{aligned}$$

3.3.2 EM algorithm for exponential family

Here, we describe the EM algorithm for exponential family following Cappé et al. (2005). Suppose that the distribution of complete-data (Z_t, Y_t) given Z_{t-1}, $p(z_t, y_t | z_{t-1})$, belongs to an exponential family, then

$$p(z_t, y_t | z_{t-1}) = h(z_t, y_t) exp\left(\langle \psi(\Theta), s(z_{t-1}, z_t, y_t) \rangle - A(\Theta)\right),$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product, $s(Z_{t-1}, Z_t, Y_t)$ is the complete-data sufficient statistic, and $\psi(\Theta)$ and $A(\Theta)$ are known functions of parameter. As a property of the exponential family, if the maximum likelihood estimate of parameter Θ exists, it will be a function of the sufficient statistics. In the case that the parameter of interest is $q(i,j)$, the sufficient statistics will be $I(Z_{t-1} = i, Z_t = j)$ where $I(\cdot,\cdot)$ is the indicator function. Assume that $S_t(i,j)$ denotes the conditional expectation of the sufficient statistics given the observation sequence $y_0, ..., y_t$ and Θ. Then, the n^{th} iteration consists of the following two steps:

- The E-step, where the expected number of transitions from state i to state j given $y_0, ..., y_t$ and $\Theta^{(n)}$, is computed,

$$S_t^{(n+1)}(i,j) = \frac{1}{t} E\left[\sum_{l=1}^{t} I(Z_{l-1} = i, Z_l = j)\middle| y_0, ..., y_t; \Theta^{(n)}\right], \qquad (7)$$

- The M-step, where the new parameter value $\boldsymbol{Q}^{(n+1)}$ is calculated using $\boldsymbol{S}_t^{(n+1)}$ and which can be formulated as $\boldsymbol{Q}^{(n+1)} = f(\boldsymbol{S}_t^{(n+1)})$ and is given by:

$$q^{(n+1)}(i,j) = \frac{S_t^{(n+1)}(i,j)}{\sum_{j=1}^{N} S_t^{(n+1)}(i,j)}. \qquad (8)$$

3.3.3 On-line EM algorithm

An advantage of the on-line algorithm, compared to the usual EM algorithm, is that the parameters are updated each time a new observation is made without the need to store the previous observations.

The on-line EM-algorithm is a modification of the EM algorithm. In order to achieve this goal, a recursive implementation of the EM algorithm will be used, see Zeitouni and Dembo (1988). The conditional expectation of the complete-data sufficient statistics, $S_t(i,j)$, can be computed recursively. To see this, define:

$$\phi_t(k) = P(Z_t = k | y_0, ..., y_t; \Theta), \qquad (9)$$

$$\rho_t(i,j,k) = \frac{1}{t} E\left[\sum_{l=1}^{t} I(Z_{l-1} = i, Z_l = j)\middle| y_0, ..., y_t, Z_t = k; \Theta\right], \qquad (10)$$

for $t = 0, 1, \ldots$. It is clear that $S_t(i,j) = \sum_{k=1}^{N} \phi_t(k)\rho_t(i,j,k)$.

We want to estimate the transition matrix $\boldsymbol{Q}$ given some observations $y_0, ..., y_t$. In n^{th} iteration of EM algorithm, all elements in $\phi_1, \phi_2, ..., \phi_t$ and $\rho_1, \rho_2, ..., \rho_t$

depend on $\boldsymbol{Q}^{(n)}$. Thus, for updating $\boldsymbol{Q}$ in $(n+1)^{th}$ iteration, all elements of the two quantities ϕ_t and ρ_t need to be recalculated. Therefore, one needs to store the entire observation vector to use the EM algorithm. The on-line EM algorithm remedies the issue of requiring the entire observation vector by using $\hat{\boldsymbol{Q}}_t$ rather than $\boldsymbol{Q}^{(t)}$. This is because one can not compute more than one iteration at each time point t for the on-line EM. As a new observation is made at time t, we update $\hat{\boldsymbol{Q}}_t$ according to Eq. (8) where $\hat{S}_t(i,j) = \sum_{k=1}^{N} \hat{\phi}_t(k)\hat{\rho}_t(i,j,k)$ and both $\hat{\phi}_t$ and $\hat{\rho}_t$ are updated recursively according to the Eqs. $(9,10)$.

To compute the auxiliary function $\hat{\rho}_t$, we use a fixed forgetting factor γ in the on-line EM algorithm. The forgetting factor specifies how quickly the algorithm forgets past information. It is needed for changing frequency of driving events over time, since for example the frequency of driving events is not the same on a highway or in a city. Cappé (2011) proposes a diminishing forgetting factor to ensure that the EM algorithm converges to a stationary point. However, this is not the goal here and we do not want the algorithm to converge to a stationary point but rather never stop adapting.

Usually, the driving conditions can change over time and a single trip may contain different road types such as city and highway. Therefore, using an on-line algorithm that adapts to a changing environment is of interest. Figure 5 shows the estimated diagonal elements of the transition matrix of driving states {RT, SF, LT} for one simulated signal. The simulated signal represents a journey on a city road, highway and then back to a city road and again highway over 10^5 seconds ($\approx$ 28 hours), where the sampling period is 1/2 seconds. The straight thick black lines show the diagonal elements of true transition matrices which are used for simulating city and highway roads.

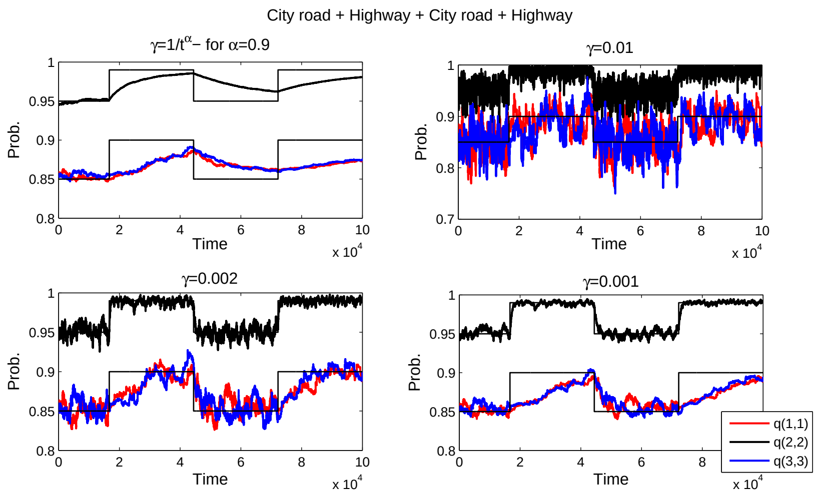


Figure 5: Diagonal elements of on-line estimated transition matrix, simulated signal from City road+Highway+City road+Highway, with four different values of forgetting factor γ. Straight thick black lines show the diagonal elements of true transition matrices Q_{city} and $Q_{highway}$.

It can be seen that the on-line algorithm with variable forgetting factor γ can not follow the changes of the parameters well and that the adaption diminishes over time, as is to be expected. The fixed forgetting factor, however, seems to adapt well to the chaining environment. If a vehicle should change the environment (city/highway) more frequently, higher values of γ would be chosen.

3.4 Estimating the number of events

Driver behaviors cause variations in the load by changing the speed, braking, and adapting to curves. The aim of this work is to develop a method for identifying the steering events and their frequency. We are mainly interested in the number of occurrences of driving events for a costumer. Together with the forces generated by the events, one can calculate the expected damage of a component for a specific vehicle. Since the steering events are not recorded in CAN data, we estimate them using a hidden Markov model (HMM), where the hidden processes represent the driving states and the observations are derived from the CAN signals.

In papers I and II, the Viterbi algorithm was used to detect the steering events. However, the Viterbi algorithm requires access to the entire data sequences and thus can not be used for on-line estimation when the data is not stored. Instead we compute the expected number of events using the on-line estimated transition matrix.

3.4.1 Detecting the driving events

The Viterbi algorithm, see Viterbi (1967), is an important algorithm in HMMs and is used to find the most probable sequence of hidden states for a new signal. Suppose that we have an observation sequence $y_0, y_2, ..., y_T$. We would like to find driving events for this new observation. It means that we should find a sequence of hidden states which maximizes the probability of observing this specified observation. The Viterbi algorithm determines the most likely sequence $\hat{z}_0, ..., \hat{z}_T$ of hidden (driving) states, which maximizes the conditional probability of the observation sequence for given parameters Θ:

$$(\hat{z}_0, ..., \hat{z}_T) = \text{argmax}_{z_0,...,z_T}\, p(y_0, ..., y_T | z_0, ..., z_T; \Theta).$$

By using the Viterbi algorithm, we reconstruct the sequence of driving states and count the number of events. Figure 6 shows a lateral acceleration signal and the corresponding reconstructed hidden states process using the Viterbi algorithm.

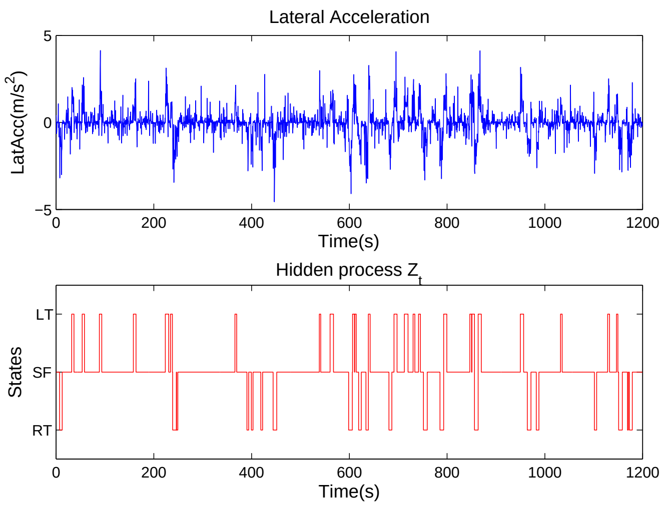


Figure 6: Lateral acceleration signal and the corresponding reconstructed hidden states.

3.4.2 Calculating the expected number of events

As mentioned above, the Viterbi algorithm requires the entire data sequences and thus can not be used when one does not store the data. Instead of detecting the driving events, one can calculate the expected number of events using a transition matrix of driving states.

Suppose that at each time t, the Markov chain $\{Z_t\}$ has transition matrix $\boldsymbol{Q}_t$. By solving the equation $(\boldsymbol{Q}_t - I)\boldsymbol{\pi_t} = \mathbf{0}$, one gets the stationary distribution of $\boldsymbol{Q}_t$. The expected number of i^{th} event for $\{Z_t\}_{t=0}^{T}$ is equivalent to the number of times that transitions $j \to i$ for all $j \neq i$ occur. The intensity of visiting state i is $\xi_i(t) = \sum_{j\neq i} I(Z_{t-1} = j, Z_t = i)$, for $t = 1, ..., T$ and $i, j = 1, 2, ..., N$. In addition, one should consider the state at time zero, Z_0, which can also be i. Therefore, the expected number of i^{th} event up to time T is:

$$\eta_i(T) = E[I(Z_0 = i)] + E[\sum_{t=1}^{T} \xi_i(t)] = \pi_{0,i} + \sum_{t=1}^{T} \sum_{j\neq i} \pi_{t,j} q_t(j, i). \tag{11}$$

In the off-line model the formula is simplified to:

$$\eta_i = \pi_i + T \sum_{j\neq i} \pi_j q(j, i).$$

4 Estimation of fatigue damage

Fatigue is a process of material deterioration caused by variable stresses. For a vehicle, stresses depend on environmental loads, e.g. road roughness, vehicle usage and driver's behavior. Time to failure of a component is a random variable that depends on many factors such as stress variability, material properties and the shape of components. Often, the so-called rainflow damage is evaluated to describe the severity of the load environments. The damage is a nondecreasing random process. In practice, the expected damage is an important parameter which is used to estimate the risks of fatigue failures. In this section we give a short introduction to the rainflow method and introduce some concepts needed to compute the expected damage for a random load.

4.1 Rainflow cycles and expected damage

The rainflow cycle count algorithm is one of the most commonly used methods to compute fatigue damage. The method was first proposed by Endo, see Matsuishi and Endo (1968). Here, we use the definition given by Rychlik (1987) which is more suitable for a statistical analysis of damage. The rainflow cycles are defined as follows.

Assume that a load L_T, the processes up to time T, has N local maxima. Let M_i denote the height of the i^{th} local maximum. Denote by m_i^+ (m_i^-) the minimum value in forward (backward) direction from the location of M_i until L_T crosses level M_i again. The rainflow minimum, m_i^{rfc}, is the maximum value of m_i^+ and m_i^-. The pair (m_i^{rfc}, M_i) is the i^{th} rainflow pair with the rainflow range $h_i(L_T) = M_i - m_i^{rfc}$. Figure 7 illustrates the definition of the rainflow cycles.

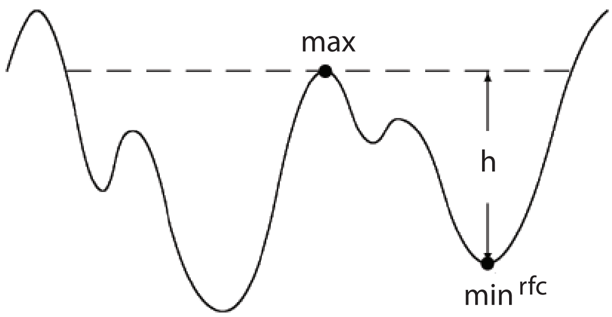


Figure 7: The rainflow cycle.

By using the rainflow cycles found in L_T, the fatigue damage can be defined by means of Palmgren-Miner (PM) rule, see Palmgren (1924), Miner (1945),

$$D_\beta(L_T) = \alpha \sum_{i=1}^{N} h_i(L_T)^\beta, \tag{12}$$

where α, β are material dependent constants. The parameter α^{-1} is equal to the predicted number of cycles with range one leading to fatigue failure (throughout the article it is assumed that α equals one). Various choices of the damage exponent β can be considered, like $\beta = 3$, which is the standard value for the crack growth process or $\beta = 5$ which is often used when a fatigue process is dominated by the crack initiation phase.

The rainflow damage, given in Eq. (12), can also be computed using the number of interval up-crossings $N^{osc}(u,v)$, viz.

$$D_\beta(L_T) = \beta(\beta-1)\int_{-\infty}^{+\infty}\int_{-\infty}^{v}(v-u)^{\beta-2}N^{osc}(u,v)\,du\,dv, \tag{13}$$

as was proved in Rychlik (1993). Note the formula is only valid for $\beta > 2$.

If L_T is a random process, one uses the expected damage as a tool to describe damage. The damage intensity of a processes is defined as:

$$d_\beta = \lim_{T\to\infty}\frac{1}{T}E[D_\beta(L_T)]. \tag{14}$$

Finally, using Eq. (13), we get that

$$d_\beta = \beta(\beta-1)\int_{-\infty}^{+\infty}\int_{-\infty}^{v}(v-u)^{\beta-2}\mu^{osc}(u,v)\,du\,dv, \tag{15}$$

where

$$\mu^{osc}(u,v) = \lim_{T\to\infty}\frac{E\left[N^{osc}(u,v)\right]}{T}. \tag{16}$$

which is called the intensity of interval up-crossings.

4.2 Reduced load and expected damage

Modelling of the external loads is an important aspect in durability studies of vehicle components. The approach taken here is to approximate the load by a vehicle independent sequence of steering events, here representing left and right turns (LT, RT) or left and right steering (SL, SR). In both cases the two events are separated by a section when wheels have approximately zero turning angle, which is called straight forward (SF). Thus, a reduced load can be defined by keeping the extreme value for each left and right turn and set to zero in between.

We describe how to construct a reduced load from steering events left and right turns. However the method could be generalized to other driving events. Let $\{Z_t\}_{t=0}^{T}$ be the hidden processes in an HMM, with three possible driving states right turn, left turn or straight forward. Assume that we have N turns. We denote the i^{th} turn by Z_i^*, which equals one if the turn is a left turn, and two if the turn is a right turn. Note that each Z_i^* corresponds to a time interval $[t_{i,start}, t_{i,stop}]$, which represents the start and stop points of i^{th} turn.

Now to create the reduced load, from the sequence of driving events, assume that M_i and m_i are the i^{th} maximum and minimum load during a turn. Then, the reduced load $\{X_i\}_{i=0}^{2N}$ is defined as follows

$$X_i = \begin{cases} 0, & \text{if } i \text{ is odd integer,} \\ M_{i/2}, & \text{if } Z^*_{i/2} = 1, i \text{ is even integer,} \\ m_{i/2}, & \text{if } Z^*_{i/2} = 2, i \text{ is even integer.} \end{cases} \tag{17}$$

Here the zeros are inserted since between each left and right turn event there must be a straight forward event. Figure 8 illustrates a lateral load and the corresponding reduced load.

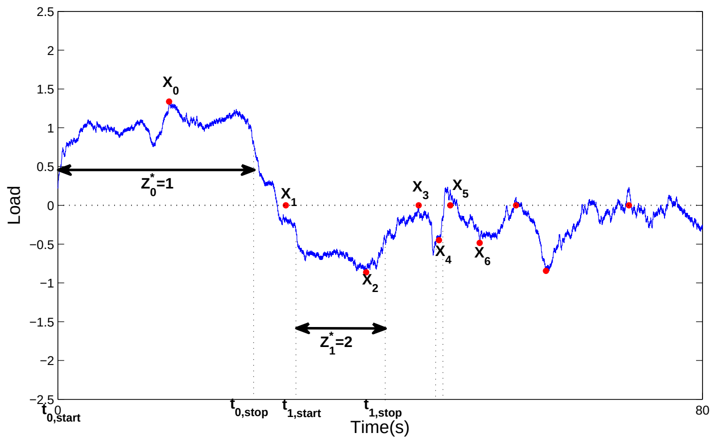


Figure 8: Reduced load represented by dots where the observed load is represented by the irregular solid line.

To compute the damage intensity d_β, per driving event, one needs the interval up-crossing intensity $\mu^{osc}(u, v)$ of random reduced load. An explicit formula for $\mu^{osc}(u, v)$ has been found for a Markov model of the reduced load. We approximate the process Z^*_i by a Markov chain with transition matrix $\boldsymbol{P} = (p(k, j))$ (it can be estimated from transition matrix $\boldsymbol{Q}$ in the HMM). Further, we assume that $\{M_i\}_{i=0}^{\infty}$ and $\{m_i\}_{i=0}^{\infty}$ are sequences of i.i.d. random variables. Then, it can be shown that the interval up-crossing intensity $\mu^{osc}(u, v)$ is:

$$\mu^{osc}(u, v) = \frac{1}{2} \begin{cases} \pi^*_2 P(m_1 < u), & u < v < 0, \\ \pi^*_2\, P(m_1 < u)\, p_2(u, v), & u \le 0 \le v, \\ \pi^*_1 P(M_1 > v), & 0 < u < v. \end{cases} \tag{18}$$

where $\boldsymbol{\pi}^* = (\pi_1^*, \pi_2^*)$ is the stationary distribution of the $\boldsymbol{P}$ and $p_2(u,v)$ can be derived from the equation system:

$$\begin{aligned} p_j(u,v) = & \, p(j,1)P(M_1 > v) + P(M_1 \leq v)\, p(j,1)\, p_1(u,v) \\ & + P(m_1 \geq u)\, p(j,2)\, p_2(u,v),\; j = 1,2. \end{aligned} \tag{19}$$

For more details see Maghsood et al. (2015).

4.3 Validation of the model

To validate the proposed load model, we compare the expected damage index with the damage index estimated from the measured data. The main steps to calculate the expected damage are as follows:

- Use the HMM to detect the steering events,
- Estimate the transition matrix $\boldsymbol{P}$,
- Estimate the distribution of the extreme forces M_i and m_i,
- Compute the interval up-crossing intensity $\mu^{osc}(u,v)$ given by Eq. (18),
- Compute the expected damage by using Eq. (15).

We use the dedicated field measurements from a Volvo Truck as our data set. Three maneuvering events are considered: steering right, steering left and straight forward. The link rod force is used as the measured load L^{obs} for damage calculation. Based on detected steering events, the reduced load x is estimated by the sequence of the extreme loads during steering right and steering left and zeros for driving straight. The Rayleigh distributions are fitted to the maximum (M_i) and minimum (m_i) values.

The measured load and the reduced load with the corresponding detected maneuvering events are shown in Figure 9a. In the figure, the stars are the extreme values of load occurring during maneuvering events and constituting the reduced load x. Figure 9b demonstrates the rainflow cycles for the measured link rod force and the reduced load. It can be seen that all large cycles can be captured by the reduced load.

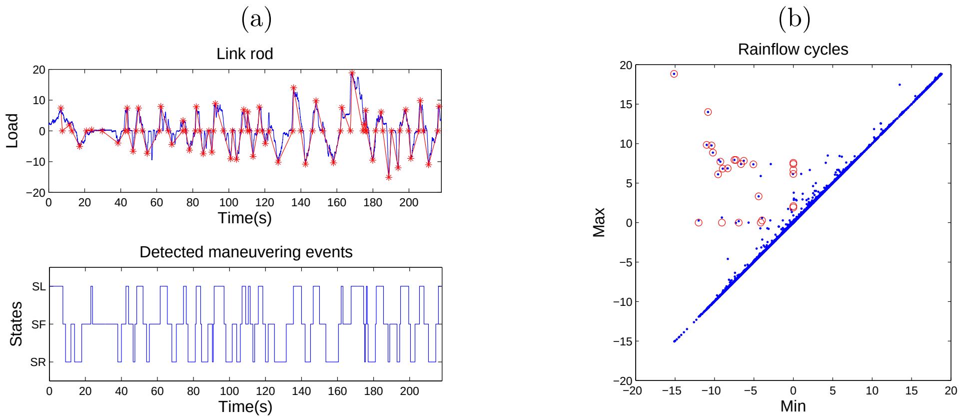


Figure 9: (a) *Top:* solid irregular line is the measured link rod force while stars represent the reduced load. *Bottom:* Detected maneuvers. (b) Dots - the rainflow cycles found in the measured link rod force. Circles - the rainflow cycles counted in the reduced load.

Table 1 shows a comparison of the damage indexes $D_\beta(L^{obs})$ computed for the measured load, $D_\beta(x)$ for the reduced load and the expected damage index $E[D_\beta(X)]$ for the random model of the reduced load. The results indicate that the damage calculated from the sequence of the extreme forces occurring during events is close to the total damage value. As expected, the damage indexes $D_\beta(L^{obs})$ and $D_\beta(x)$ are almost identical. We conclude that the reduced load models the variability of the measured load well. The expected damage $E[D_\beta(X)]$ is larger than $D_\beta(x)$. Whether this difference is significant is shown in Figure 10.

Table 1: Comparison of damage indexes $D_\beta(L^{obs})$ computed for the measured load, $D_\beta(x)$ for the reduced load and the expected damage index $E[D_\beta(X)]$.

	$D_\beta(L^{obs})$	$D_\beta(x)$	$E[D_\beta(X)]$
$\beta = 3$	$14.8 \cdot 10^3$	$14.3 \cdot 10^3$	$15.5 \cdot 10^3$
$\beta = 5$	$2.3 \cdot 10^6$	$2.2 \cdot 10^6$	$2.7 \cdot 10^6$

In Figure 10a, the load spectra estimated from the measured link rod force and the reduced load are compared with the theoretical load spectrum. As can be seen in Figure 10b, where the load spectra for 10 simulated loads are compared with the theoretical load spectrum and the load spectrum of the reduced load, the differences between the measured spectrum and the expected one are small.

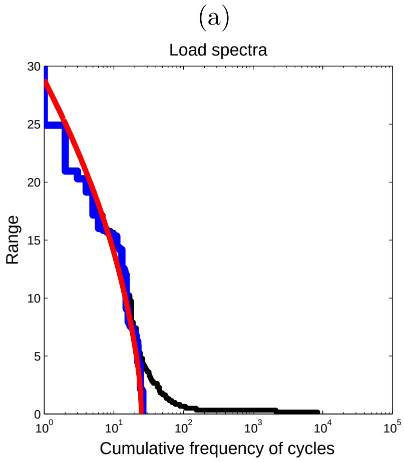


Figure 10: (a) The regular solid line is the theoretical load spectrum. The stair-like functions are the load spectra found in measured link rod force and the reduced load. (b) Load spectra for 10 simulated loads compared with the theoretical load spectrum and the load spectrum of the reduced load (the thick stair-like line).

4.3.1 Damage investigation based on on-line algorithm

In this section we compute the damage intensity per kilometer based on on-line estimation of transition matrix . We use one simulated lateral acceleration signal in order to calculate the damage. The simulated signal represents a journey during a city road, highway and then back to a city road and again highway over 10^5 seconds, where the sampling period is 1/2 seconds. The speed of the vehicle is considered to be 50 kilometers per hour and the mileage is 1000 km. We split the signal into 1000 equally sized frames. For each frame, the expected number of turns are computed by $\Delta\eta_k = \eta_k - \eta_{k-1}$, where η_k is the estimated number of turns occurring over the first k kilometers. The expected damage based on turns for each frame is calculated by:

$$\Delta d_k = \Delta\eta_k d_k,$$

where d_k is the expected damage per turn and calculated by means of Eqs. (15) and (18). The empirical distribution of M_i and m_i is used to calculate the intensity of interval crossings $\mu^{osc}(u, v)$. We use the on-line estimation of transition matrix $\boldsymbol{Q}$ with forgetting factor $\gamma = 0.002$ to estimate the transition matrix $\boldsymbol{P}$. The result for damage exponent $\beta = 3$ is shown in Figure 11. The straight thick red line shows $\Delta d_k(\boldsymbol{Q}_{true})$, which is the damage intensity computed using the model transition matrices for city and highway. We can observe the change in damage between highway and city road. As might be expected the damage intensities (per km) estimated for the city are higher than for highway, since the number of turns occurring in a city road is larger than on highways.

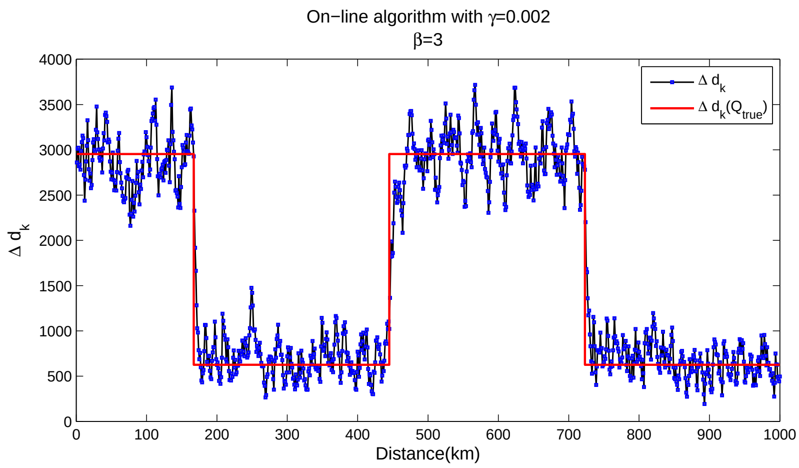


Figure 11: Damage intensity per km according to the on-line estimation of transition matrix with $\gamma = 0.002$. The plot shows the results for damage exponent $\beta = 3$. The straight thick red line shows $\Delta d_k(\boldsymbol{Q}_{true})$, which is the damage intensity computed using model transition matrices for city and highway.

5 Summary of papers

5.1 Paper I: Detection of Steering Events based on Vehicle Logging Data using Hidden Markov Models

Hidden Markov models (HMMs) have been proposed for detection of steering events such as curves and maneuvering using on-board logging signals available on trucks, such as lateral acceleration, vehicle speed and steering wheel angle. The idea is to consider the current driving event as the hidden state and construct the model based on them. To estimate the parameters of the model, we have considered 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 differences between roads can affect transitions between states. The emission matrix describes the property of the events given certain hidden states, while the transition matrix describes the sequence and duration of the events. Therefore, it could be relevant to update the transition matrix for a new signal to find the hidden states.

In order to compare methods 1 and 2, 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 not detect the true event, the false negative error will happen. 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.

5.2 Paper II: Modeling Extreme Loads Acting on Steering Components using Driving Events

In this paper the damage index is calculated based on identified driving events such as curves and maneuvers. The sequence of the most extreme forces occurring during driving events will form the largest cycles and are consequently the most essential parts of the load. We have modeled the sequence of driving events with a Markov chain and reduced load by keeping the extreme value for each left and right turn and zero for the straight forward event. A formula for the expected damage index is computed using reduced load.

The proposed models were validated using measured data from a Volvo truck. The results show that all large rainflow cycles found in measured load were found in the reduced load. Hence, the fatigue damage of steering components can be predicted by reduced load. The proposed load model accurately describes the variability of the rainflow ranges for the considered measured load. Further, using the model the expected damage could be predicted.

5.3 Paper III: Load Description and Damage Evaluation using Vehicle Independent Driving Events

In this paper, we investigated the main parameters of the load model proposed in Paper II. The proposed random model depends only on four parameters. A sensitivity study was conducted to see how much the expected damage depends on the variability of parameters of the proposed model.

5.4 Paper IV: Detection of Steering Events using Hidden Markov Models with Multivariate Observations.

In this paper, we have proposed a method to identify the steering events, such as curves and maneuvers, using an HMM with multivariate observation sequences. We considered six driving states: right turn (RT), straight forward for turns (SFT), left turn (LT), steering right (SR), straight forward for maneuvers (SFM) and steering left (SL), to construct the HMM. It is shown that hidden Markov models with a combination of continuous and discrete distributions for observations can be used to find the steering events. Both simulated and measured signals are used to identify the steering events and validate the model.

5.5 Paper V: On-line Estimation of Driving Events and Fatigue Damage on Vehicles

In this article, we estimated the number of steering events using a hidden Markov model where the model parameters are estimated by using an on-line EM algorithm. An advantage, compared to the usual EM algorithm, is that the parameters are re-estimated each time as a new observation is made without the need to store the previous observations. The driving conditions can change over time and a single trip may contain different road types such as city and highway. Therefore, we propose an algorithm that adapts to a changing environment. Further, we compute the expected damage caused by driving events using an on-line algorithm.

By determining the number of events that occur in customer vehicles, it is possible to estimate the fatigue damage caused by the same kind of events. We have shown how to compute the expected damage according to the reduced load using an on-line estimation of transition matrix.

6 Conclusion

It has been found that the HMMs can be used to recognize the steering events such as curves and maneuvers based on vehicle logging data. We estimate the steering events using an HMM, where the hidden process is the driving states and the observations are CAN signals. We use the EM algorithm to estimate the HMM parameters. The parameters of the HMM are also estimated using an on-line EM algorithm. The on-line algorithm is a modification of the EM algorithm which allows for an adaptive parameter estimation method. By using an on-line algorithm, the parameters can adapt over chaining driving environment.

Damage calculation is investigated for steering components. The sequence of the most extreme forces occurring during driving events will form the largest cycles and are consequently the most essential part of the load. The sequence of driving events which is modeled by a Markov chain, forms the vehicle independent part of the load. The load is approximated using the vehicle independent sequence of driving events, here representing left and right turns or left and right steering. In both cases the two events are separated by a section when wheels have zero turning angle, which is called Straight forward. Thus, a reduced load can be defined by keeping the extreme value for each left and right turn and zero between each event. An explicit formula is found for the expected damage according to the reduced load. We conclude that the proposed random load accurately describes the variability of the rainflow ranges for the measured loads in question.

References

Barndorff-Nielsen, O., Kent, J., and Sorensen, M. (1982). Normal variance-mean mixtures and z distributions. *International Statistical Review*, 50:145–159.

Berndt, H. and Dietmayer, K. (2009). Driver intention inference with vehicle onboard sensors. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 102–107, Pune.

Bogsjö, K., Podgorski, K., and Rychlik, I. (2012). Models for road surface roughness. *Vehicle System Dynamics*, 50:725–747.

Cappé, O. (2011). Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*, 20:3:728–749.

Cappé, O., Moulines, E., and Rydén, T., editors (2005). *Inference in Hidden Markov Models*. Springer.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer.

Karlsson, M. (2007). *Load Modelling for Fatigue Assessment of Vehicles – a Statistical Approach*. PhD thesis, Chalmers University of Technology, Sweden.

Kotz, S., Kozubowski, T., and Podgorski, K. (2001). *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media.

Kvanström, M., Podgórski, K., and Rychlik., I. (2013). Laplace moving average model for multi-axial responses in fatigue analysis of a cultivator. *Probabilistic Engineering Mechanics*, 34:12–25.

Maghsood, R., Rychlik, I., and Wallin, J. (2015). Modeling extreme loads acting on steering components using driving events. *Probabilistic Engineering Mechanics*, 41:1320.

Matsuishi, M. and Endo, T. (1968). Fatigue of metals subjected to varying stress. *Japan Society of Mechanical Engineers. In Japanese.*

Miner, M. A. (1945). Cumulative damage in fatigue. *Journal of Applied Mechanics*, 12:A159–A164.

Mitrović, D. (2004). *Learning Driving Patterns to Support Navigation.* PhD thesis, University of Canterbury, New Zealand.

Mitrović, D. (2005). Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):198–205.

Nilsson, T., Nyberg, P., Sundström, C., Frisk, E., and Krysander, M. (2014). Robust driving pattern detection and identification with a wheel loader application. *International Journal of Vehicle Systems Modelling and Testing*, 9(1):56–76.

Palmgren, A. (1924). Die Lebensdauer von Kugellagern. *Zeitschrift des Vereins Deutscher Ingenieure*, 68:339–341. *In German.*

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rychlik, I. (1987). A new definition of the rainflow cycle counting method. *International Journal of Fatigue*, 9:119–121.

Rychlik, I. (1993). Note on cycle counts in irregular loads. *Fatigue & Fracture of Engineering Materials & Structures*, 16:377–390.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13(2):260–269.

Zeitouni, O. and Dembo, A. (1988). Exact filters for the estimation of the number of transitions of finite-state continuous-time Markov processes. *IEEE Transactions on Information Theory*, 34(4):890–893.

Paper I

Detection of Steering Events based on Vehicle Logging Data using Hidden Markov Models

R. Maghsood and P. Johannesson

Detection of steering events based on vehicle logging data using hidden Markov models

Roza Maghsood*

Mathematical Sciences,
Chalmers University of Technology,
SE-412 96 Gothenburg, Sweden
Email: rozam@chalmers.se
*Corresponding author

Pär Johannesson

SP Technical Research Institute of Sweden,
P.O. Box 24036, SE-400 22 Gothenburg, Sweden
Email: Par.Johannesson@sp.se

Abstract: In vehicle design it is desirable to model the loads by describing load environment, customer usage and vehicle dynamics. In this study a method will be proposed for detection of steering events such as curves and manoeuvring using on-board logging signals available on trucks. The method is based on hidden Markov models (HMMs), which are probabilistic models that can be used to recognise patterns in time series data. In an HMM, 'hidden' refers to a Markov chain where the states are not observable. However, observations depending on the hidden Markov chain can be observed. The idea here is to consider the current driving event as the hidden state, while the on-board logging signals generate the observed sequence. Examples of curve detection are presented for both simulated and measured data on a truck. The classification results indicate that the method can recognise left and right turns with small misclassification errors.

Keywords: HMMs; hidden Markov models; Markov chain; Viterbi algorithm; Baum-Welch algorithm; steering events; event classification; on-board logging signals; lateral acceleration.

Reference to this paper should be made as follows: Maghsood, R. and Johannesson, P. (2016) 'Detection of steering events based on vehicle logging data using hidden Markov models', *Int. J. Vehicle Design*, Vol. 70, No. 3, pp.278–295.

Biographical notes: Roza Maghsood is a PhD student in Mathematical Statistics at the Chalmers University of Technology. She was awarded her MSc in Mathematical Statistics from the Allameh Tabatabaei University in Iran.

Pär Johannesson received his PhD in Mathematical Statistics in 1999 at the Lund Institute of Technology, with a thesis on statistical load analysis for fatigue. He has published 10 papers in international journals, and is Co-Editor of the handbook "Guide to Load Analysis for Durability in Vehicle Engineering". During 2000–2001, he had a PostDoc at Mathematical Statistics at Chalmers and PSA Peugeot Citroën in Paris, and from 2002 to 2010 he was a Researcher at the Fraunhofer-Chalmers Centre in Gothenburg. He is currently a Researcher at the

SP Technical Research Institute of Sweden, working with statistical methods for load analysis, reliability and fatigue.

This paper is a revised and expanded version of a paper entitled 'Detection of the curves based on lateral acceleration using hidden Markov models' presented at *5th Edition of the International Conference on Fatigue Design*, Senlis, France, 27–28 November, 2013.

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, see Edlund and Fryk (2004) for heavy truck applications, Karlsson (2004, 2007) for curve characteristics, and Johannesson and Speckert (2013) for general information. 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, customer vehicles in general have no access to measurements dedicated to durability, and using specially equipped vehicles in service is difficult and expensive. Thus, for on-board logging of events we need to use the information which is available for all vehicles by means of controller area network (CAN) bus data.

For a typical heavy truck, there are more than 80 signals from the CAN bus and some of them, such as steering wheel angles, vehicle speed, yaw rate and lateral acceleration, are really important to identify the driving events. The steering wheel is the wheel turned by the driver while driving and the angle is defined as the angle deviation from driving straight ahead. The yaw rate is the angular velocity. The lateral acceleration is proportional to the lateral force in turning events. For most components these loads are not as damaging as the vertical loads, but they have a large impact on steering components (Karlsson, 2007).

A dataset 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 and their frequencies from CAN data. In this study we propose a method using HMMs to detect steering events such as curves and manoeuvring based on vehicle logging data. The idea is to use the driving events, i.e., straight forward and turns, as the hidden states and construct an HMM based on them.

The HMMs have been widely used in signal processing to recognise the events and also to predict them in the future, see e.g., the overview by Rabiner (1989) with applications to speech recognition. Mitrović (2004, 2005) and Berndt and Dietmayer (2009) 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 in roundabouts. 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 characterise 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 easier to estimate the parameters of one model than a large number 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 steering events such as the start and stop points.

Lateral acceleration signals have been used to detect the curves. We have simulated different lateral acceleration signals with different lengths and different number of curves (events) to have some controlled training and test sets. We have also used the same technique to detect the manoeuvres such as static steering. The measured steering wheel angle signal has been used instead of lateral acceleration.

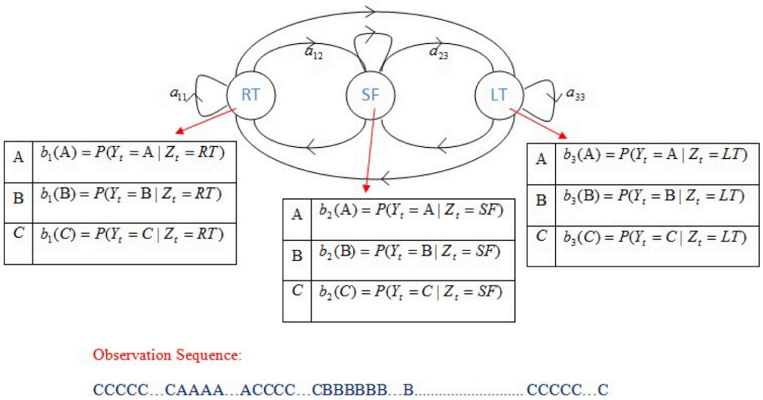
In Section 2, we describe the concept of HMMs and present two methods for detection of curves. For method 1 the parameters of the HMM are estimated from the training set, while for method 2 the transition matrix is re-estimated based on the test set. Examples with results for simulated and measured data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

2 Hidden Markov models

Hidden Markov models are probabilistic models 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, which 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 depend 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.

First, 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, the transitions between them 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 (see online version for colours)



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

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

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

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

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

and $\sum_{k=1}^{M} b_j(V_k) = 1$.

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

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

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

It has been demonstrated that a discrete HMM can be good in pattern recognition, see Rabiner (1989). 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 above, we have three hidden states $S = \{$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 containing observations 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, $V = \{$A, B, C$\}$, which are defined as follows:

- A $= \left\{\text{“lateral acceleration”} < -0.2 \text{ m/s}^2\right\}$
- B $= \left\{-0.2 \text{ m/s}^2 \leq \text{“lateral acceleration”} \leq 0.2 \text{ m/s}^2\right\}$
- C $= \left\{\text{“lateral acceleration”} > 0.2 \text{ m/s}^2\right\}$.

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 normalised 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 recognise the curves. The parameters of the HMM have been estimated by using a training set and evaluated 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 follows:

- *Method 1*: Use the estimated transition and emission matrices from the training set to detect events in the test set.
- *Method 2*: Use the emission matrix from the training set but re-estimate the transition matrix based on the test set, and then detect events in the test set.

The main reason for considering method 2 is that differences between types of roads can affect the transition probabilities 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 (1967) and Forney (1973), has then 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 have the observation sequence $y_1, y_2, \ldots, y_n$. We would like to find driving events for this new observation. It means that we want to find a sequence of hidden states which maximises the probability of making this specified observation. The Viterbi algorithm finds the state sequence $z_1, z_2, \ldots, z_n$ out of the 3^n possible sequences of length n, which maximises:

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

In fact, the Viterbi algorithm gives a state sequence $z_1, z_2, \ldots, z_n$, which maximises 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 for each new signal. To estimate model parameters based on an observation sequence when the hidden sequence is unknown, we have used the *Baum-Welch* algorithm, which was introduced by Baum et al. (1970). It is a special case of the expectation-maximisation (EM) algorithm, see Dempster et al. (1977). The Baum-Welch algorithm is one of the most well-known methods for estimating the model parameters in HMMs on unlabelled sequences. 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.

For completeness, we will describe the Baum-Welch algorithm following Rabiner (1989), and then we will explain how it has been used in method 2.

Consider an observation sequence $y_1, y_2, \ldots, y_T$ with length T and suppose that we have N hidden states $S = \{S_1, S_2, \ldots, S_N\}$. We will 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:

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

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:

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

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 $t = 1, 2, \ldots, T-1$.

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 will compute $\xi_t(i,j)$. Based on the definition of forward-backward factors, $\xi_t(i,j)$ can be written as follows:

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

  where $\sum_{t=1}^{T} \xi_t(i,j)$ is the expected number of transitions from state i to j. Let's define $\gamma_t(i) = P(Z_t = S_i | Y_1 = y_1, \ldots, Y_T = y_T; \lambda)$. Then it should 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 the M-step, the parameters $\lambda = (A, B, \pi)$ will be updated. Both forward and backward variables will be used to re-estimate model parameters by using the following formulas:

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

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

3 Examples

We have tested our models with simulated and measured datasets. 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

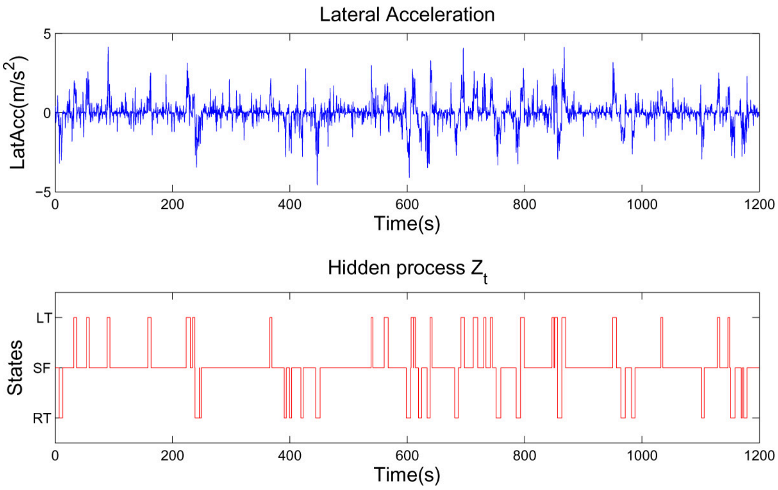
We need a training set to estimate the parameters and a test set to evaluate the model. For this purpose, we have simulated different lateral acceleration signals with a sampling period of 0.5 s. Figure 2 shows an example of the simulated lateral acceleration signal and the corresponding hidden states. These two simulated signals will be our training set. Next, we will describe how the simulation has been performed.

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 for the sequence of events such as

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

where $p = 0.1$ and represents 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.

Figure 2 Simulated lateral acceleration signal and the corresponding hidden states (see online version for colours)



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$(i), i = 1, \ldots, K$ 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 2–8 s
- 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. Note that the constructed Z_t is not a Markov chain, since the lengths of the curves are uniformly distributed, which is more realistic than the exponential case.

To generate a lateral acceleration signal, we have used a model suggested by Karlsson (2006, 2007). 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, $a_{\text{trap}}(t)$ the centripetal acceleration, and $a_{\text{res}}(t)$ the residual, which is a combination of different factors such as driver, road, vehicle and velocity. The model is formulated as:

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

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 $a_{\text{trap},j} = v_j^2 \times C_j$ where v_j indicates the constant speed over curve j and C_j is the maximum curvature. Further, the curvature is modelled by introducing the transformation:

$$Y_{C_j} = 1/C_j - r_{\mathrm{turn}},$$

where $\log Y_{C_j} \sim \mathrm{Normal}(\mu, \sigma^2)$, 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 (2007), 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 4.

3.1.1 Estimate of parameters from training set

It should be remembered 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 s (2400 samples) of the training set.

Even though the simulated hidden process is not a Markov chain, we will use the HMM methodology to estimate parameters and to detect events. 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*:

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

- *The emission matrix*:

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

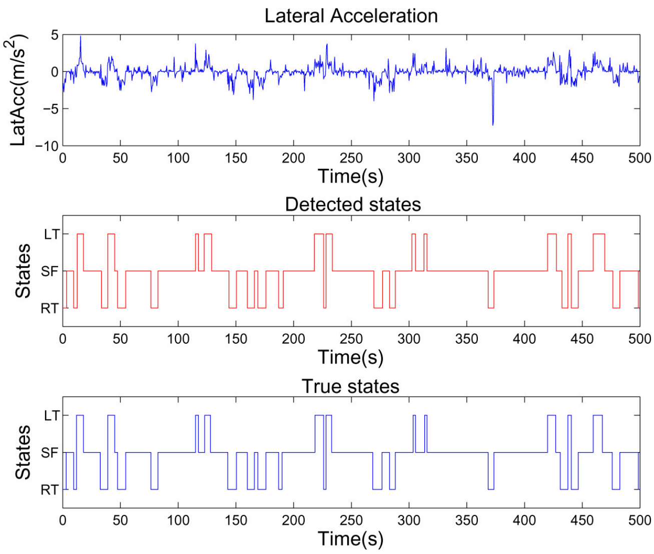
3.1.2 Model evaluation

To recognise 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 example, the method can recognise left turn and right turn without any misclassification errors.

Figure 3 Detection of events using method 1 (see online version for colours)



Method 2

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

The re-estimated transition matrix:

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

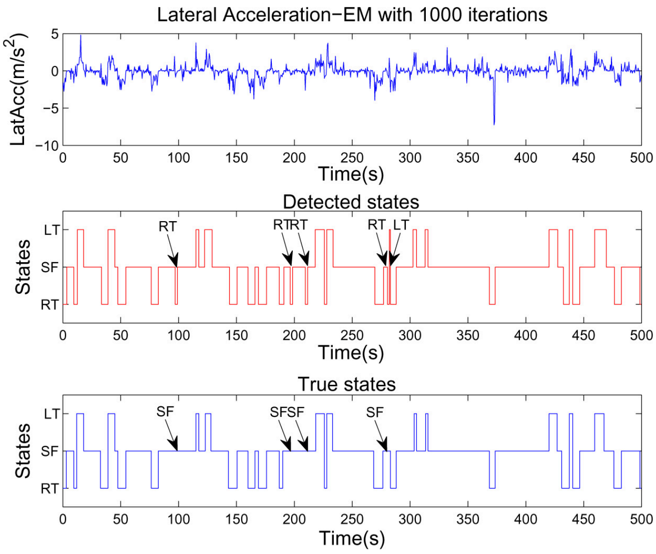
The true and detected states for the new signal are shown in Figure 4, where we can see that there are some misclassifications, and hence the misclassification error rate in this case is higher than in method 1.

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 cannot detect the true event, the false negative error will happen.

Most of the time, the duration of the detected events is 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.

Figure 4 Detection of events using method 2 (see online version for colours)



We have simulated signals with different numbers of events as our test sets ($n = 10, 100$), in order to investigate the sensitivity to the sample size. We have changed the parameters for the test sets to check how much they affect the results. For each case, we performed 1000 simulations to get an average of error rates. Three cases of straights of different duration have been considered by setting $\theta = 4, 20, 100$. For instance, by using $\theta = 100$ we can illustrate a highway. Note that the length of the simulated signal depends both on n and θ. The modified transition probabilities for the Markov chain is:

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

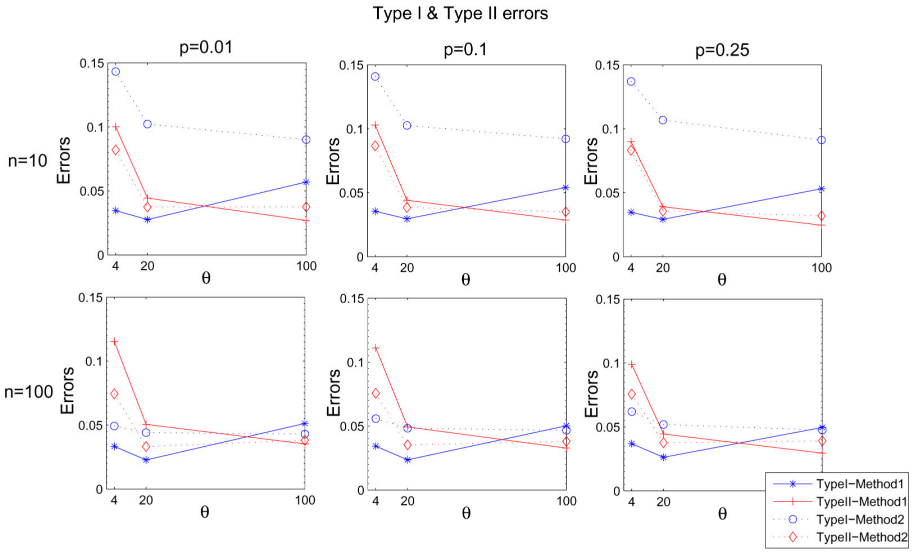
where the value $p = 0.01, 0.1, 0.25$. If we have a larger value of p it means we have more curves in our simulations.

Figure 5 shows the error rates with regard to the different parameters. It can be seen that if we change the p value then the figures will not change very much but if we change the value of θ then there is quite a great 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 preferred since we do not have enough data to estimate the parameters.

The simulation study does not give any clear observations, but it gives 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. 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 cannot be good and it should be re-estimated from the new signal. Hence, method 2 should be used.

Figure 5 Type I and Type II errors (see online version for colours)



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 is 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 for detection of events and compared its result with the HMM detection. The aim is to demonstrate the benefit of the HMM approach compared to a simple threshold-based method.

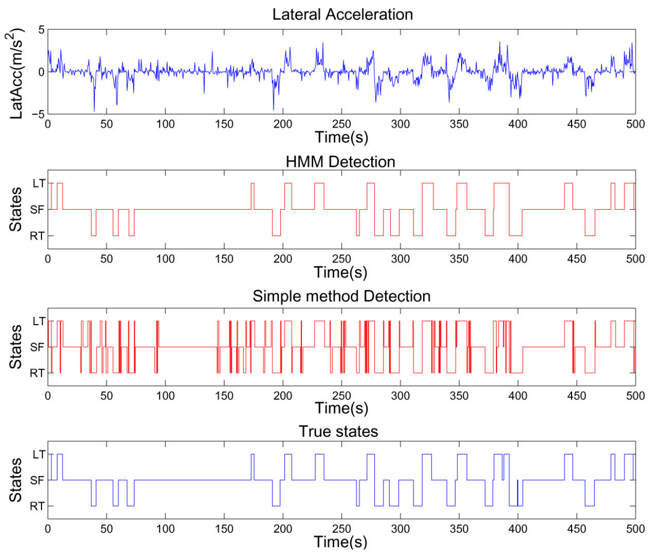
The method detects the curves by using pre-defined thresholds for lateral acceleration signals. If the absolute value of lateral acceleration is larger than 0.2 m/s^2 for more than 1.5 s, then the algorithm will detect the event as a turning event.

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 work much better than the simple algorithm to recognise the turning events.

3.3 Measured lateral acceleration signal

The measured data that we have used are field measurements coming from a Volvo Truck. We have used the 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 happened during the driving. By having the start and stop points of each event, we have created the hidden Z-process.

Figure 6 Comparing HMMs with a simple method (see online version for colours)



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

$$\text{“lateral acceleration”} = \text{“speed”} \cdot \text{“yaw rate”}.$$

To remove the high frequency noise, we have used a low-pass filter with 0.5 Hz cut-off frequency. To reduce the amount of data, we have split 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

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

Compared to the earlier clustering, 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 of 3800 s, which we have divided into two parts as our training and test sets. The training set has a duration of 2000 s and the test set 1800 s. Figure 7 shows the training part of the signal and the corresponding manually detected hidden states.

Method 1

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

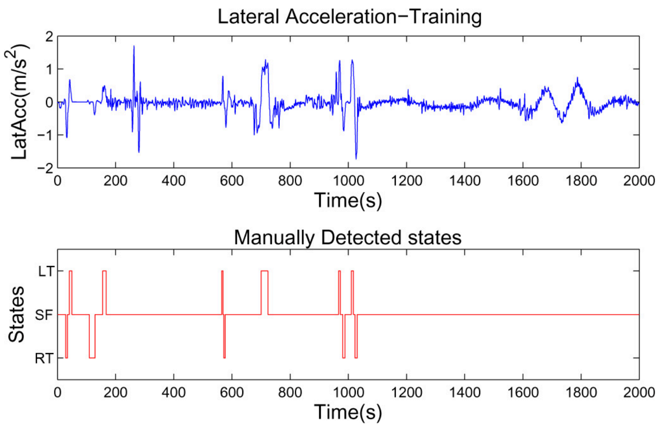
$$\begin{array}{cc} & \begin{array}{ccc} R & S & L \end{array} \\ \begin{array}{c} R \\ S \\ L \end{array} & \begin{pmatrix} 0.945 & 0.055 & 0 \\ 0.001 & 0.997 & 0.002 \\ 0 & 0.048 & 0.952 \end{pmatrix} \end{array}$$

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 are shown in Figure 8, where we can compare them with the manually detected states.

Figure 7 Training part of measured lateral acceleration signal and the corresponding manually detected hidden states (see online version for colours)



It can be seen that the misclassification error rate is quite high. In all cases, the method can recognise 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 used for the manual detection. There is also a sharp left turn at the end which could not be detected since the speed is low (about 10 km/h), thus giving low lateral acceleration, and making it hard to recognise the curve correctly. However, this last curve is on the borderline between curve and manoeuvring events, see below.

Method 2

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

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

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, methods 1 and 2 performed similarly to detect the events.

Figure 8 Detection of events in the measured signal using method 1 (see online version for colours)

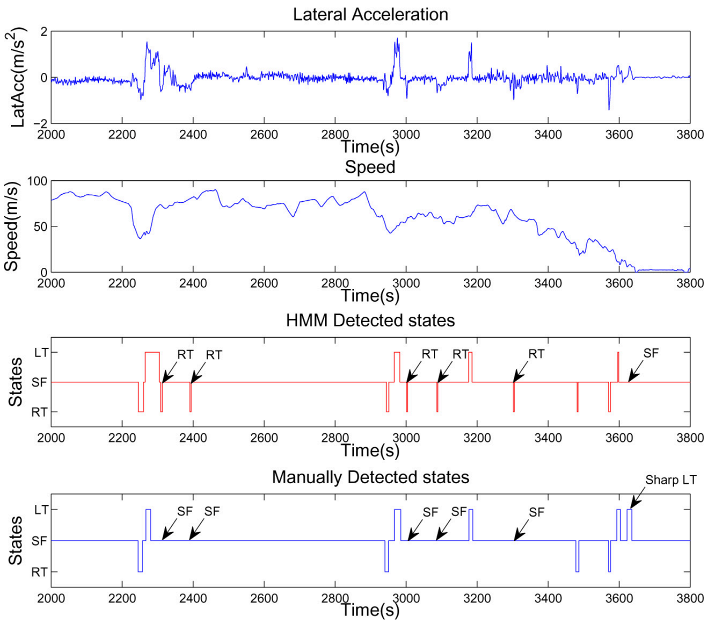
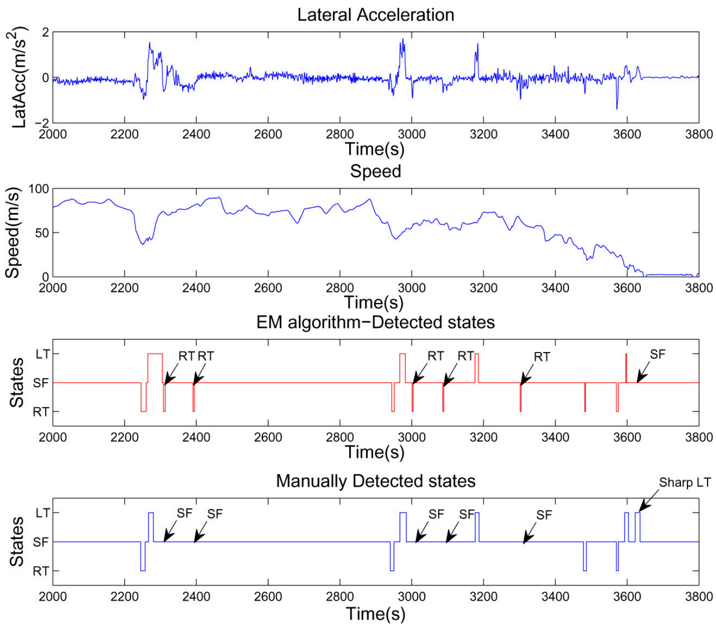


Figure 9 Detection of events in the measured signal using method 2 (see online version for colours)



3.4 Detection of manoeuvring events

For many steering components the so-called manoeuvring events generate the highest forces. The manoeuvring events are considered as the events which will happen at speed less than about 10 km/h. For instance, driving out of the parking lot and turning to the right and left, reversing and manoeuvring into the parking lot and standing still but turning the steering wheel are typical manoeuvring events.

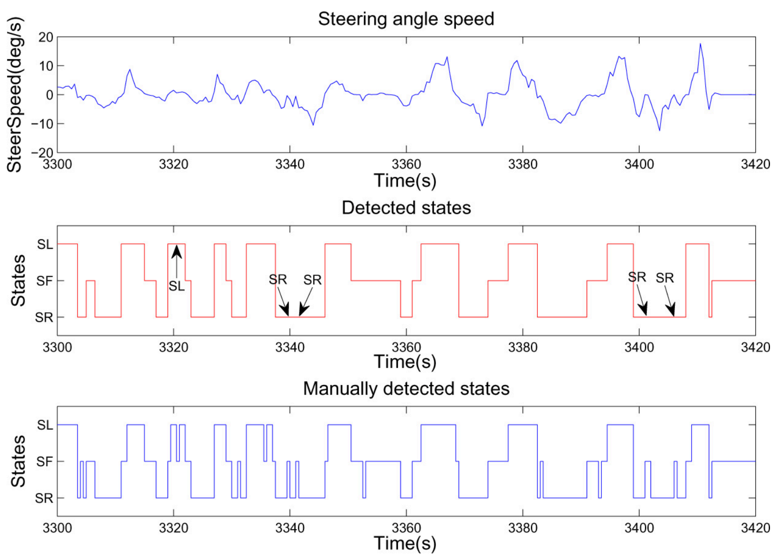
We have applied the same technique as for the curves to detect manoeuvring events by using HMMs. Three events *steering right* (SR), *steering left* (SL) and *straight forward* (SF) have been considered as our hidden states (process Z_t). We have considered the steering angle speed signal as our data and translated this continuous feature into predefined classes (process Y_t). Three new classes have been used as follows:

- A $= \{$"Steering angle speed" < -0.75 deg/s$\}$,
- B $= \{-0.75$ deg/s $\leq$ "Steering angle speed" ≤ 0.75 deg/s$\}$,
- C $= \{$"Steering angle speed" > 0.75 deg/s$\}$.

where the threshold 0.75 deg/s has been chosen based on experience.

Figure 10 shows the detected events based on HMMs and the corresponding manually detected hidden states for a measured signal from the CAN bus data. For the estimation of the HMM parameters, we have used method 1, i.e., the parameters are estimated based on the manual detection as the training set. It can be seen that the HMM detection of events agrees quite well with the manual detection. As mentioned before, the manual detections are not completely correct because of the visual errors and the low quality of videos.

Figure 10 Detection of steering events in the measured signal (see online version for colours)



4 Conclusion

The examples in this study indicate that the HMMs can be used to recognise steering events such as curves and manoeuvring based on vehicle logging data. We have considered the steering events, i.e., 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 are 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. For instance, 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 cannot be good and it should be re-estimated from the new signal. In that case method 2 should be used.

In this study, we have separated the curve and manoeuvring events and constructed two different models, HMM_C for the curves and HMM_M for the manoeuvres. Another approach can be to combine the two models into a larger HMM containing both curve and manoeuvring events. In that case there will be several essential signals which will increase the number of classes in the Y-process.

Acknowledgment

We are thankful to Professor Igor Rychlik, Dr. Thomas Svensson and Dr. Marina Axelson-Fisk for their 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

Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970) 'A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains', *The Annals of Mathematical Statistics*, Vol. 41, No. 1, pp.164–171.

Berndt, H. and Dietmayer, K. (2009) 'Driver intention inference with vehicle onboard sensors', *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Pune, pp.102–107.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) 'Maximum likelihood from incomplete data via EM algorithm', *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39, No. 1, pp.1–38.

Edlund, S. and Fryk, P. (2004) *The Right Truck for the Job with Global Truck Application Descriptions*, SAE Technical Paper. 2004-01-2645.

Forney, G.D. (1973) 'The Viterbi Algorithm', *Proceedings of the IEEE*, Vol. 61, pp.268–278.

Johannesson, P. and Speckert, M. (Eds.) (2013) *Guide to Load Analysis for Durability in Vehicle Engineering*, Wiley, Chichester.

Karlsson, M. (2004) 'Driver independent road curve characterisation', *Supplement to Vehicle System Dynamics*, Vol. 41, pp.411–420.

Karlsson, M. (2006) 'Statistical parameterization of lateral vehicle loads', *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, Vol. 44, Supp. 1, pp.770–779.

Karlsson, M. (2007) *Load Modelling for Fatigue Assessment of Vehicles – A Statistical Approach*, PhD Thesis, Chalmers University of Technology, Sweden.

Mitrović, D. (2004) *Learning Driving Patterns to Support Navigation*, PhD Thesis, University of Canterbury, New Zealand.

Mitrović, D. (2005) 'Reliable method for driving events recognition', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, No. 2, pp.198–205.

Rabiner, L.R. (1989) 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proceedings of the IEEE*, Vol. 77, No. 2, pp.257–286.

Viterbi, A.J. (1967) 'Error bounds for convolutional codes and an asymptotically optimal decoding algorithm', *IEEE Transactions on Information Theory*, Vol. IT-13, pp.260–269.

Paper II

Modeling Extreme Loads Acting on Steering Components using Driving Events

R. Maghsood, I. Rychlik and J. Wallin

Probabilistic Engineering Mechanics 41 (2015) 13–20

Contents lists available at ScienceDirect

Probabilistic Engineering Mechanics

journal homepage: www.elsevier.com/locate/probengmech

Modeling extreme loads acting on steering components using driving events

R. Maghsood*, I. Rychlik, J. Wallin

Mathematical Sciences, Chalmers University of Technology, SE-412 96 Göteborg, Sweden

ARTICLE INFO

Article history:
Received 18 March 2015
Accepted 10 April 2015
Available online 18 April 2015

Keywords:
Fatigue damage index
Hidden Markov models (HMMs)
Laplace distribution
Rainflow cycles
Vehicle independent load models
Steering events

ABSTRACT

Forces during steering events, such as curves and maneuvers, cause large stresses on steering components. In this paper, we formulate a model for the lateral loads causing fatigue damage of the steering components. Steering events are identified using a Hidden Markov model on the CAN (Controller Area Network) bus data. The CAN data is available on all vehicles, thus the model is applicable across many types of vehicles. To identify the events, the observation from CAN data is modeled by a multivariate generalized Laplace (GAL) distribution. An explicit formula for the expected fatigue damage is given. Results are validated using measured lateral acceleration.

1. Introduction

Fatigue is a process of material deterioration caused by variable stresses which may lead to failures of metallic components. Only parts of loads causing large oscillations influence the fatigue life. Knowledge about the variability of external loads can be used to design market specific vehicles. In this study we consider steering components for which the large stress oscillations occur during turnings which will be called *driving events*.

The paper deals with two problems: The first is development of methods to estimate some market specific statistical characteristics of driving events, e.g. the frequency of events. The second is computation of the expected damage for a steering component.

There exist a large number of different markets and customer types for which one would like to know the load acting on vehicles. Field measuring of forces is both time consuming and costly. Thus it is proposed to use statistics of occurrences of driving events to describe load environments encountered by vehicles. In this paper we present means to estimate driving events statistics using CAN (Controller Area Network) bus data, which is available for all vehicles. Since the driving events are not recorded in CAN data we extract them using a hidden Markov model (HMM) [6], where each event represents a hidden state.

The idea of using HMMs to find driving events is not new, see Maghsood and Johannesson [18,17], Mitrović [21,22] and [1]. The novelty of our approach is to use a multivariate generalized Laplace model (GAL) as the distribution of the observation process in the HMM. Recent results show that Laplace models are well suited to describe responses measured on driving vehicles, see [27,16,4]. In previous works, the observation process has been discretized by using thresholds. The continuous model, used here, has two main advantages over the discrete version. First, in the discrete model one has to manually set the threshold levels, whereas the continuous model is entirely estimated from the actual data. Second, the continuous model can easily be extended to incorporate multivariate sources of information, which is not easy for the discrete threshold approach.

Using the hidden states above, we get a sequence of driving events, which we model with a Markov chain (MC). The MC is specific to the market or customer type, and since it is vehicle independent we use it to describe the variability of expected load environment. However, fatigue life depends on vehicle responses and properties of the steering component used. Consequently given a model for load environment, one needs to evaluate expected damage for a component. This is done as follows: using laboratory tests or designed measurements on test trucks, a distribution of loads acting on a component during a driving event is found. By attaching independent random variables, with the event specific distributions derived above, on each steering event, we have constructed a reduced load model. By the Markov property for the reduced load model a closed form formula for the expected damage is derived. Modeling load sequences by Markov chains is not new, see e.g. [3,8–11,13,15,25,28]. However, the presented formula was not given previously in the literature and the stringent proof of the result is presented in the Appendix.

* Corresponding author.
E-mail address: rozam@chalmers.se (R. Maghsood).

http://dx.doi.org/10.1016/j.probengmech.2015.04.002
0266-8920/© 2015 Published by Elsevier Ltd.

Finally, we validate our method on a real data set, with field measurements of loads acting on a steering component in a Volvo truck. First, we show that the steering events found in CAN data are responsible for the large oscillation of forces acting on steering component. This is done by constructing a reduced load, which consists only of the extreme load values occurring during steering events. If the damage from the reduced load is close to the observed damage computed using continuously measured forces, we conclude that the steering events accurately capture the variability of the load. Second, we verify that the expected damage of the reduced random load is close to the observed damage.

The paper is organized as follows. The theory and some preliminaries are presented in Section 2. The proposed load model and the way for detecting driving events are presented in Section 3. In Section 4, measured data are used to validate the models and to illustrate the results.

2. Theory and preliminaries

In this section, theory and known results needed in this paper are presented. We start with a short description of HMMs and the generalized asymmetric Laplace distribution used to detect the sequence of driving events in CAN data. Further counting rainflow cycles procedure is presented in Section 2.2, while definition of damage index is given in Section 2.3.

2.1. Event identification using HMMs with Laplace distribution

Hidden Markov models [6] are one of the most useful statistical models to identify the patterns in signals. The model consists of two processes: a hidden, that is not observed, Markov chain $\{S_t\}_{t=0}^{\infty}$ and a observed process $\{Y_t\}_{t=0}^{\infty}$. Conditioning on $\{S_t\}_{t=0}^{\infty}$, the observed process $\{Y_t\}_{t=0}^{\infty}$ is a sequence of independent random variables with distributions depending only on S_t.

In this paper, Y_t has the generalized asymmetric Laplace distribution (GAL), see [14]. As mentioned in the Introduction the Laplace distribution has recently been successfully used to describe the loads acting on vehicles.

The GAL distribution is defined by the following parameters: δ – location vector, μ – shift vector, $\nu > 0$ – shape parameter, and Σ – scaling matrix and denoted by $GAL(\delta, \mu, \nu, \Sigma)$. The one-dimensional GAL distributions dependences on the shape parameter, ν, is illustrated in Fig. 1. An important property of the $GAL(\delta, \mu, \nu, \Sigma)$ distribution is that it has an explicit formula for the probability density function (pdf), namely

$$f(y) = \frac{1}{\Gamma(\lambda)\sqrt{2\pi}} \left(\frac{\sqrt{(y-\delta)^T \Sigma^{-1} (y-\delta)}}{c_2} \right)^{(\nu - d/2)/2} e^{(y-\delta)\Sigma^{-1}\mu}$$
$$K_{\nu - d/2}\left(c_2 \sqrt{(y-\delta)^T \Sigma^{-1}(y-\delta)} \right),$$

where d is the dimension of Y, $c_2 = \sqrt{2 + \mu^T \Sigma^{-1} \mu}$ and $K_{\nu-d/2}(.)$ is the modified Bessel function of the second kind. A convenient representation of a random variable Y having GAL distribution is

$$Y = \delta + \Gamma\mu + \sqrt{\Gamma}\,\Sigma^{1/2} Z,$$

where Γ is the Gamma distributed with shape ν and scale one, while Z is a vector of d independent standard normal random variables.

2.2. Rainflow cycle

The rainflow cycle count algorithm is one of the most commonly used methods to count cycles. The method was first proposed by Matsuishi and Endo [19]. Here, we shall use the definition given in [24] which is more suitable for statistical analysis of damage index. Assume that a load x has N local maxima. Let M_i denote the height of ith local maximum. Denote by m_i^+ (m_i^-) the minimum value in forward (backward) direction from the location of M_i until x crosses M_i again. The rainflow minimum, m_i^{rfc}, is the maximum value of m_i^+ and m_i^-. The pair (m_i^{rfc}, M_i) is the ith rainflow pair and $h_i(x) = M_i - m_i^{rfc}$. Fig. 2 illustrates the definition of the rainflow cycles.

Counting rainflow cycles is equivalent to counting the number of interval upcrossings by a load, denoted by $N^{osc}(u, v)$, see [26,2] for multivalued loads.

Remark 1. Note that some local maxima cannot be paired with any of local minima in x. It will happen when the corresponding rainflow minimum m_i^{rfc} lies before or after the period that load was measured. The sequence of maxima and minima which could not be paired by means of rainflow method is called the residual and has to be handled separately. Here, we let maxima in the residual form cycles with the preceding minima in the residual.

2.3. Fatigue damage index

The most common way to define fatigue damage, using rainflow cycles, is the Palmgren–Miner (PM) rule [23,20]

$$D_\beta(x) = \alpha \sum_{i=1}^{N} h_i(x)^\beta, \tag{1}$$

where α, β are material dependent constants. The parameter α^{-1}

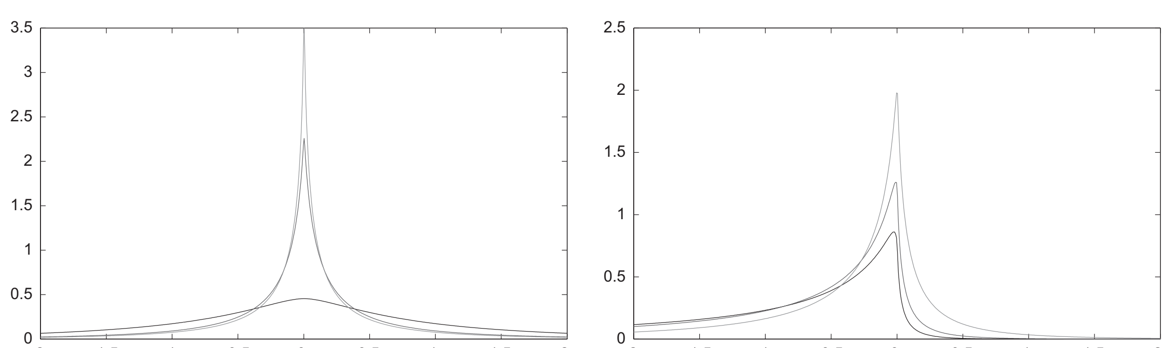


Fig. 1. The figures illustrates the one-dimensional GAL pdf dependence of the shape parameter ν. For the figure to the *left* the parameter μ is zero giving symmetrical pdf, while to the *right* μ is negative.

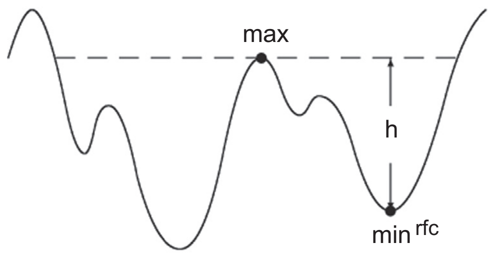


Fig. 2. The rainflow cycle.

is equal to the predicted number of cycles with range one leading to fatigue failure (throughout the paper it is assumed that α equals one). Various choices of the damage exponent β can be considered, like $\beta = 3$ which is the standard value for the crack growth process or $\beta = 5$ which is often used when a fatigue process is dominated by the crack initiation phase.

The rainflow damage, given in Eq. (1), can also be computed using number of interval upcrossings $N^{osc}(u, v)$, viz.

$$D_\beta(x) = \beta(\beta-1)\int_{-\infty}^{+\infty}\int_{-\infty}^{v}(v-u)^{\beta-2}N^{osc}(u, v)\,du\,dv, \tag{2}$$

as was proved in [26]. Note that the formula is only valid for $\beta > 2$.

For a discrete random load $X = (X_1, \ldots, X_n)$, the damage $D_\beta(X)$ is a random variable. The average growth of the expected damage

$$d_\beta = \lim_{n\to\infty}\frac{1}{n}E[D_\beta(X)]. \tag{3}$$

is an important parameter describing severity of the random load. The average d_β will be called the damage index. Note that due to nonlinearity of rainflow counting method one has that

$$E[D_\beta(X)] \le n\cdot d_\beta.$$

Finally, using Eq. (2), we get that

$$d_\beta = \beta(\beta-1)\int_{-\infty}^{+\infty}\int_{-\infty}^{v}(v-u)^{\beta-2}\mu^{osc}(u, v)\,du\,dv, \tag{4}$$

where

$$\mu^{osc}(u, v) = \lim_{n\to\infty}\frac{E[N^{osc}(u, v)]}{n}, \tag{5}$$

which is called the intensity of interval upcrossings. Note that $N^{osc}(u, v)$ is a function of X and hence also of n.

3. Reduced load

Let $x(t)$ denote a load acting on a steering component. In order to define the reduced load, first suitable driving events causing large load oscillations need to be defined. In this paper steering events are right turn, left turn and straight forward (RT, LT, ST). The reduced load $\{x_i\}_{i=0}^{N}$ consists of the extremal loads during the turns.

More precisely, denote the ith turn by Z_i which equals one if the turn is a left turn, and two if the turn is a right turn. Note that each Z_i corresponds to a time interval $[t_{i,start}, t_{i,stop}]$, which represents the start and stop points of ith turn.

Next, let

$$M_i = \max_{t\in[t_{i,start}, t_{i,stop}]} x(t), \qquad m_i = \min_{t\in[t_{i,start}, t_{i,stop}]} x(t),$$

i.e. the ith maximum and minimum load during a turn. Further, it is assumed that a vehicle is driving straight forward (SF) between two turns and we let the reduced load be zero. Thus, the reduced

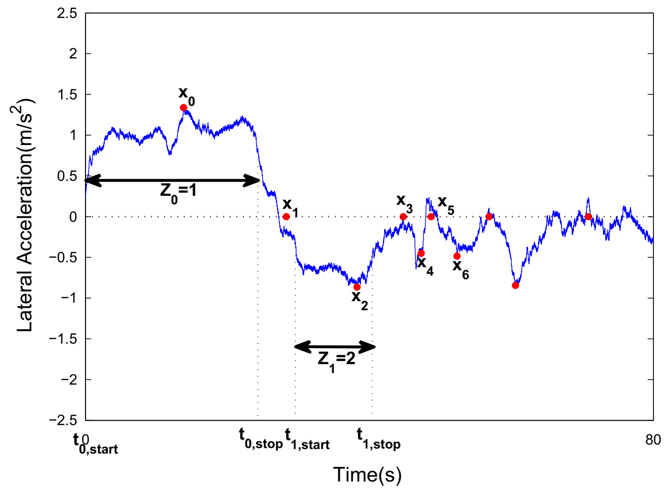


Fig. 3. Reduced load x represented by dots compared with observed load x^{obs} (lateral acceleration) represented by the irregular solid line.

load is defined as follows:

$$x_i = \begin{cases} 0 & \text{if } i \text{ is odd integer,} \\ M_{i/2} & \text{if } Z_i = 1,\ i \text{ is even integer,} \\ m_{i/2} & \text{if } Z_i = 2,\ i \text{ is even integer.} \end{cases} \tag{6}$$

Finally, Fig. 3 illustrates a measured lateral load $x^{obs}(t)$, say, and the corresponding reduced load x_i.

3.1. Markov model for reduced load

Environmental loads acting on vehicles often vary in an unpredictable way. This property can be modeled by means of random processes, i.e. one assumes that the measured load $x(t)$ is a realization of a random load $X(t)$. Consequently the reduced load defined in (6), denoted now by $\{X_i\}_{i=0}^{\infty}$, is a sequence of random variables.

Here, we approximate the process Z_i in (6) by a Markov chain with two states, 1:="LT", 2:="RT", and transition matrix $P = (p_{kj})$. Further, assume that $\{M_i\}_{i=0}^{\infty}$ and $\{m_i\}_{i=0}^{\infty}$ are independent and identically distributed random variables. (M_i and m_i, which are vehicle dependent, may have different distributions.) The reduced random load X_i is now defined by means of (6).

3.2. Damage index for reduced load

For random reduced load $X = (X_0, X_1, \ldots)$ the damage index d_β, defined in (3), is given by $d_\beta = \lim_{n\to\infty}\frac{1}{n}E[D_\beta(X)]$. The damage index can be computed using (4) whenever the interval upcrossing intensity $\mu^{osc}(u, v)$ is known. In this section we give an explicit formula for $\mu^{osc}(u, v)$ for a Markov model of the reduced load. In order to evaluate the intensity $\mu^{osc}(u, v)$ one first need to compute probabilities $p_i(u, v)$, $i = 1, 2$, where $p_i(u, v)$ is the conditional probability that given $Z_0 = i$, the sequence of extreme loads will visit the set $(v, +\infty)$ before it visits $(-\infty, u)$. It will be shown in the Appendix that the probabilities satisfy the following equation system:

$$p_i(u, v) = p_{i1}P(M_1 > v) + P(M_1 \le v)p_{i1}p_1 + P(m_1 \ge u)p_{i2}p_2, \quad i = 1, 2. \tag{7}$$

An explicit formula for the interval upcrossing intensity by X_i is given in the following theorem.

Theorem 2. *For X_i defined in Eq. (6), $\mu^{osc}(u, v)$ is given by*

$$\mu^{osc}(u, v) = \frac{1}{2}\begin{cases} \pi_2 P(m_1 < u), & u < v < 0, \\ \pi_2 P(m_1 < u) p_2(u, v), & u \le 0 \le v, \\ \pi_1 P(M_1 > v), & 0 < u < v, \end{cases} \tag{8}$$

where $p_2(u, v)$ is a solution to the equation system in (7).

Proof of the theorem is given in the Appendix.

3.3. Event reconstruction

In Section 3, Eq. (6), a reduced load was defined. In order to create $\{x_i\}_{i=0}^{N}$ one needs to find the number of driving events N, the variables $t_{i,start}$, $t_{i,stop}$ and Z_i, for $i = 1, \ldots, N$. In this section, we will discuss the challenging task of finding sequence Z_i in a, possible multivariate, signal $\{y_t\}_{t=1}^{T}$. In this paper the signal $\{y_t\}_{t=1}^{T}$ comes from CAN data.

We start by formulating an HMM with multivariate GAL distribution on the signal $\{y_t\}_{t=1}^{T}$. We denote the hidden states $\{S_t\}_{t=1}^{T}$, which represent the driving events, LT, RT, and SF.

The model has two different types of parameters that need to be estimated. The first type is the parameters in the distribution of y_t given S_t, which for our model is GAL. In this paper, these parameters are found by maximum Likelihood estimation on a training data set where the driving events are known (such information is not included in CAN data). The second type is the transition matrix of $\{S_t\}_{t=1}^{T}$, which is obtained from the signal y_t using an expectation maximization (EM) algorithm, see [7,6]. Using the Viterbi algorithm, see Viterbi [29], we recover the most likely sequence, $\hat{S}_t$, of events given the data and the estimated parameters.

The elements Z_i used in (6) is created as follows: First, let $\{t_k\}_{k=1}^{N*}$ denote the indices when $\hat{S}_t$ switches its values that are the indices where $\hat{S}_{t_k} \neq \hat{S}_{t_k-1}$. Further, for $k = 1, \ldots, N*$, define

$$Z_k^* = \begin{cases} 0 & \text{if } S_{t_k} = SF, \\ 1 & \text{if } S_{t_k} = LT, \\ 2 & \text{if } S_{t_k} = RT. \end{cases}$$

Finally, the sequence of events $\{Z_i\}_{i=1}^{N}$ is defined as $\{Z_k^*\}_{k=1}^{N*}$ with the zero elements removed. The time intervals $(t_{i,start}, t_{i,stop})$ equal (t_k, t_{k+1}) where k is the index of Z_i in $\{Z_k^*\}_{k=1}^{N*}$.

4. Example

In this section, we study lateral acceleration loads on a Volvo truck. The lateral load is known to cause damage on steering components. The measurements come from CAN data, and to construct the lateral acceleration signal, we plug the yaw rate and speed from CAN data into the following formula:

$$x = \frac{\text{speed}\cdot\text{yaw rate}}{3.6}. \tag{9}$$

The latent events used are steering events occurring when vehicle is driving with speed higher than 10 km/h, e.g. when driving in curves. Three events right turn (RT), left turn (LT) and straight forward (SF) are assessed in this study. From these event we construct a reduced load, and then calculate the damage index.

The lateral acceleration signal from CAN data is proportional to the lateral loads and is used to identify the steering events and damage calculation.

It is important to point out that typically the related load acting on steering components is not the same as the signal used for detection. For instance, the maneuvering events, e.g. driving in or out of a parking lot, standing still but turning steering wheel, generate the highest forces in steering components. To detect the maneuvers, the steering angle speed from CAN data is used while for damage calculation the link rod force is used. The link rod force is not included in the CAN data and it has been separately measured. The model proposed in this paper was also used in this case and the derived reduced model gave very accurate estimates of the damage index. Unfortunately, this example will not be presented in this paper, since the data set was very small and thus we could not use it for a rigorous statistical analysis.

4.1. Detection of steering events using HMMs

In order to find the steering events we set a HMM model on the lateral acceleration. A training set is used to estimate the parameters of HMM and a test set is used to validate the model. In our study, the training set contains all necessary information about the events such as the duration of the curves, whereas for test set we use only the lateral acceleration.

We manually identified the events by looking at video recordings from the truck cabin to create the training and test sets. By having the start and stop points of each event, we have created the sequence of steering events. We split up the data into training and testing sets. For the training set, we fitted the generalized Laplace (GAL) distribution for observations within each state. This gives us the conditional densities of the observed lateral acceleration for each steering events. The fitted distributions for lateral acceleration values within each event are shown in Fig. 4.

On the test data we get, using the EM-algorithm, the estimated transition matrix:

$$\begin{array}{c c} & \begin{array}{ccc} \text{RT} & \text{SF} & \text{LT} \end{array} \\ \begin{array}{c} \text{RT} \\ \text{SF} \\ \text{LT} \end{array} & \begin{pmatrix} 0.948 & 0.049 & 0.003 \\ 0.003 & 0.994 & 0.002 \\ 0.002 & 0.032 & 0.966 \end{pmatrix} \end{array}$$

Finally, we detected 34 turns in the signal using Viterbi algorithm. The measured lateral acceleration signal and the detected steering events based on HMM for the test set are shown in Fig. 5, where we can compare the detection result with the manually identified events.

It can be seen that the HMM algorithm can recognize most of the manually detected turns. However, the manually identifications are not completely correct because of the visual errors and the low quality of videos.

4.2. Validation of reduced load

In this section we investigate if the random load approximates the measured load x^{obs} sufficiently well. Based on the detected steering events, the reduced load $x = (x_0, \ldots, x_n)$ is estimated by the sequence of the extreme loads during right and left turns and zeros for driving straight.

To evaluate the accuracy of the proposed load model, two issues should be investigated:

(I) Whether the reduced load x contains all large rainflow cycles that were found in the load x^{obs}. By this we control whether the assumption that load is zero when the vehicle is driving straight forward is not too crude and whether the HMM algorithm detects correctly the steering events.

(II) Whether the random load $X = (X_0, \ldots, X_n)$, defined in Section 3.1 using Eq. (6), is accurately describing the variability of rainflow ranges counted in the reduced load x.

We investigated the problem (I) by comparing the rainflow cycles found in the measured load x^{obs} and in the reduced load x.

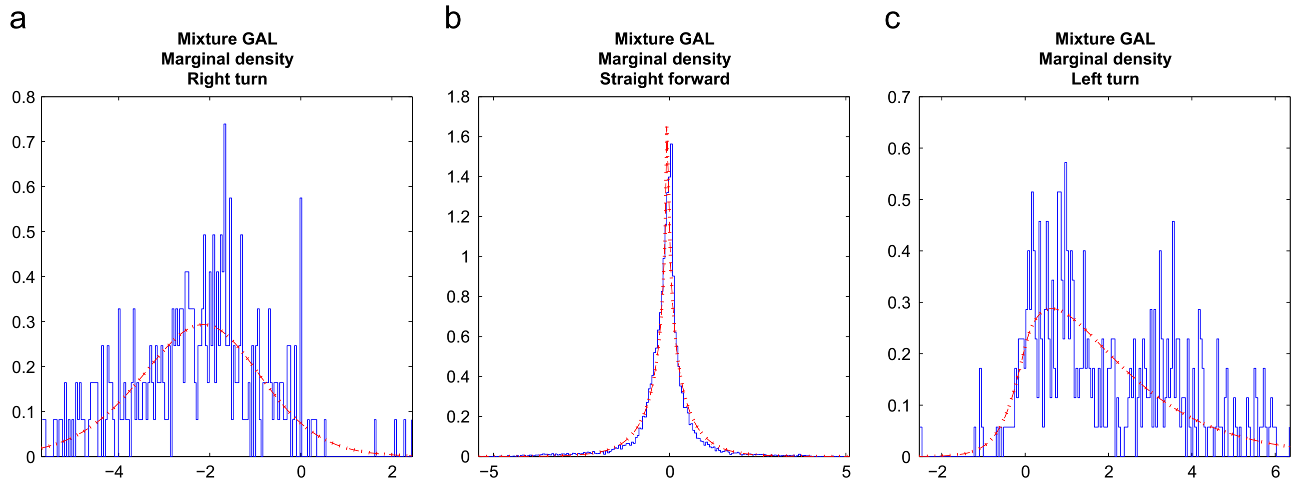


Fig. 4. (a), (b) and (c) represent the Laplace distributions fitted on lateral acceleration values for right turns, straight forward and left turns respectively.

Furthermore the damages $D_\beta(x^{obs})$, $D_\beta(x)$ and the expected damage $E[D_\beta(X)]$, for $\beta = 3, 5$, are evaluated.

The problem (II) is addressed by studying the variability of the damage $D_\beta(X)$ and checking whether $D_\beta(x)$ does not differ significantly from samples of $D_\beta(X)$. In addition the rainflow range spectrum, see Eq. (11), found in x is compared with the expected spectrum and with the simulated spectra, i.e. found in samples of X.

4.2.1. Comparison of rainflow counts

The sequence of detected right and left turns are modeled by a Markov chain. The transition matrix P is estimated, by counting the number of transitions between the turns, to

$$\begin{pmatrix} 0.32 & 0.68 \\ 0.86 & 0.14 \end{pmatrix}.$$

The extreme values of load occurring during the steering events are modeled by independent Rayleigh distributed variables. The Rayleigh distributions are fitted to the maximum value of x^{obs} during a left turn and minimum value of x^{obs} during a right turn giving

$$P(M_1 > v) = e^{-(1/2)(v/0.8)^2}, \quad v \ge 0, \qquad P(m_1 < u) = e^{-(1/2)(u/0.7)^2}, \quad u \le 0. \tag{10}$$

The matrix P and the parameters of Rayleigh distributions define together the reduced random load X.

As mentioned above, the lateral acceleration is used as the load x^{obs}. The load is shown in the top plot of Fig. 6(a) as a solid irregular line. In the figure, the stars are the extreme values of load occurring during right and left turns and constituting the reduced load x. Rainflow cycles have been found both in the load and in the reduced load and are compared in Fig. 6(b). The rainflow cycles found in the measured load are marked as dots having coordinates (m^{rfc}, M). One can see that there are few large cycles and many small ones. The rainflow cycles found in the reduced load are

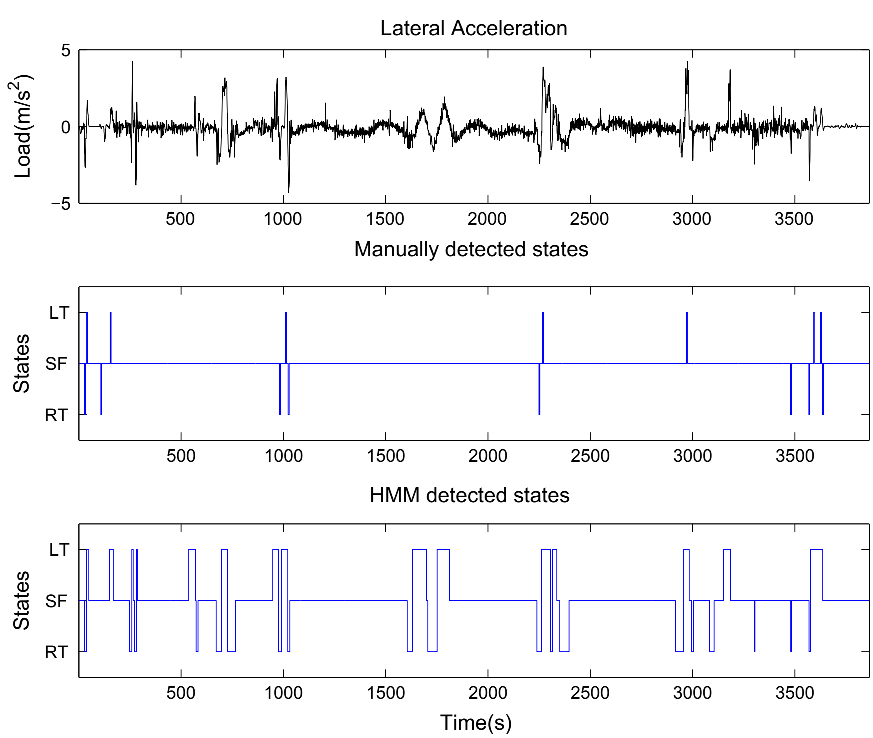


Fig. 5. *Top*: lateral acceleration signal. *Middle*: manually detected steering events. *Bottom*: detected events based on HMM.

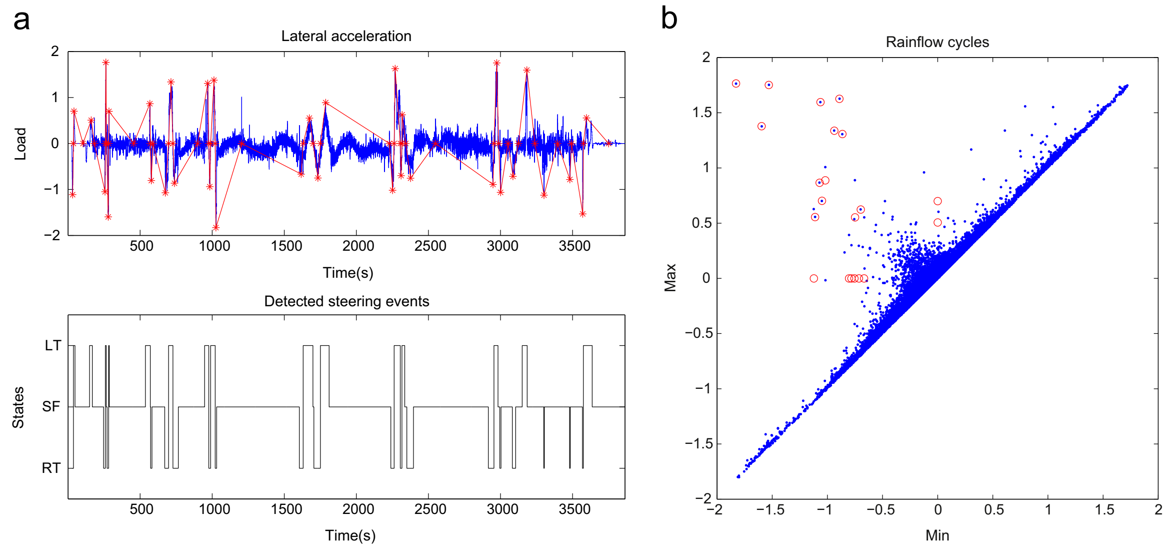


Fig. 6. (a) *Top*: solid irregular line is x^{obs} (lateral acceleration) while the thin red line is x (reduced load). Stars represent the extreme values of load occurring during the steering events. *Bottom:* detected curves. (b) Dots – the rainflow cycles found in x^{obs}. Circles – the rainflow cycles counted in x. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

presented as circles. As can be seen in Fig. 6(b), the largest cycles are counted both in lateral acceleration and in the reduced load. However there are also many moderate size rainflow cycles found in the lateral acceleration which are missing in the reduced load rainflow count. We conclude that the largest cycles found in both the lateral acceleration signal and the reduced load are not of higher order of magnitude than cycles occurring during the event SF. Hence, we can expect that the damage computed for the lateral acceleration signal will be higher than the damage estimated for the reduced load.

The damages evaluated for the load (lateral acceleration), reduced load and the expected damage are compared in Table 1. The numerical integration in (4) as well as the rainflow cycle counting has been done using the WAFO (Wave Analysis for Fatigue and Oceanography) toolbox, see [5,30,31], which can be downloaded free of charge.

Since the largest rainflow cycles are found both in lateral acceleration and reduced load, the damages $D_5(x^{obs})$ and $D_5(x)$ are almost identical. The large number of moderate size cycles found when the vehicle was driving straight forward is contributing to $D_3(x^{obs})$ and not to $D_3(x)$ and hence there is larger difference between values of these two damages. The expected damage $E[D_\beta(X)]$ is larger than $D_\beta(x)$. Whether this difference is significant will be investigated next.

4.2.2. Comparison of load spectra

We investigated whether the variability of the reduced load x is well modeled by the random load X. It is well known that the cycles with small ranges do not contribute much to the fatigue damage and hence their distribution not need to be accurately modeled. However these may heavily influence the ranges cdf leading to rejection of practically "good" model. In engineering

Table 1
Comparison of damages $D_\beta(x^{obs})$ computed for the measured load, $D_\beta(x)$ for the reduced load and the expected damage $E[D_\beta(X)]$.

	$D_\beta(x^{obs})$	$D_\beta(x)$	$E[D_\beta(X)]$
$\beta = 3$	33.6	24.7	26.9
$\beta = 5$	53.7	51.6	79.7

one often prefers to use the so-called load spectra to compare rainflow cycles distributions. The load spectrum is defined as follows:

Consider a load y having N rainflow cycles and the rainflow ranges with the cdf $F^{rfc}(h)$. Let H be a random variable having cdf $F^{rfc}(h)$. Then, the damage is

$$D_\beta(y) = N{\cdot}E[H^\beta] = \beta N{\cdot}\int_0^\infty P(H > h)h^{\beta-1}\,dh.$$

If for two loads the functions $N{\cdot}P(H > h) = N(1 - F^{rfc}(h))$ are close for high and moderate values of h, then for any $\beta > 1$ the damage indices are close too. Traditionally, one defines the load spectrum $S(h)$ to be the inverse of function of $N(1 - F^{rfc}(h))$. Then, the plot of load spectrum $(h, S(h))$ coincides with the graph of the following line:

$$\left(N(1 - F^{rfc}(h)), h\right), \quad h \geq 0, \tag{11}$$

see [12] for more details.

The load spectrum in (11) was found for the lateral acceleration x^{obs} and reduced load x. The expected load spectrum was also evaluated by integrating $n{\cdot}\mu^{osc}(u,v)$ over suitable regions. In Fig. 7 (a), the load spectra for the lateral acceleration, reduced load x and the random load X are compared. The observed load spectrum contains much more small and moderately high ranges than the remaining two spectra. Further the expected load spectrum, shown as a smooth line, is close to the load spectra of the reduced load.

In Fig. 7(b), the expected load spectrum is compared with 10 load spectra computed from simulated samples of the random load. In the figure, the smooth solid line is the expected load spectrum evaluated using $n{\cdot}\mu^{osc}(u,v)$ while the thick stairs looking like line is the load spectrum found in the reduced load. Except cycles with very small ranges, one can see that the load spectrum of x does not differ significantly from the simulated load spectra.

5. Conclusion

In this paper, we have proposed an hidden Markov model (HMM) for detection of steering events using on-board logging signals available on trucks. Using the model we have shown how

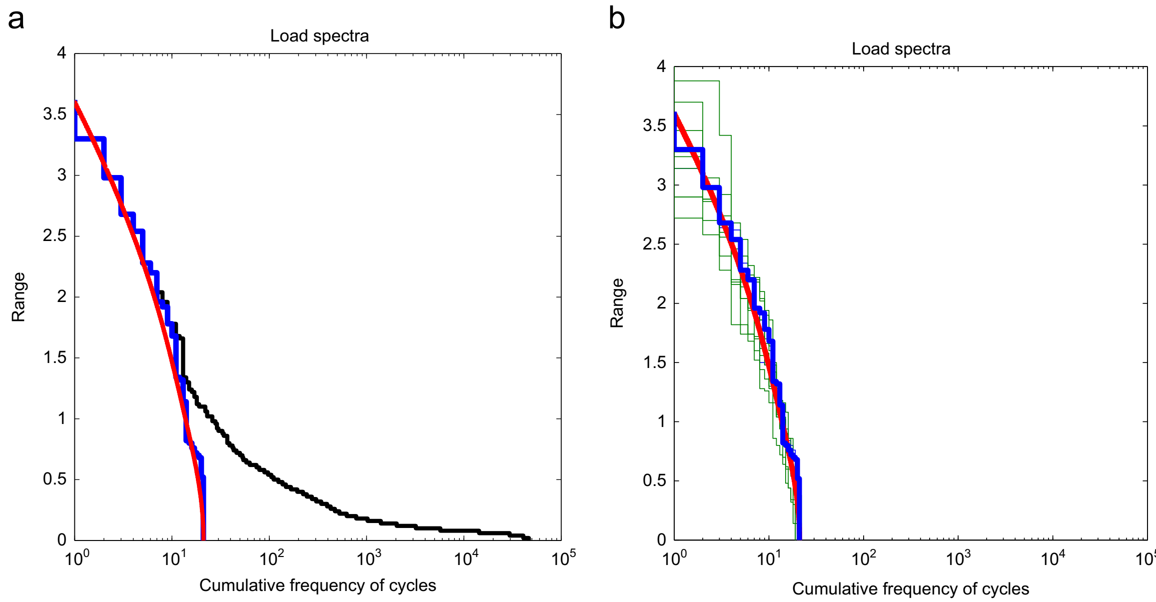


Fig. 7. (a) Comparison of load spectra (rainflow ranges). The regular solid line is the theoretical spectrum computed from $n\cdot\mu^{osc}(u,v)$, $n=68$. The stairs like functions are the observed load spectra in measured load (lateral acceleration (9)) and the reduced load (b). Comparison of load spectra found in simulations of random load X, see (6), with theoretical load spectrum and the load spectrum of the reduced load (the thick stairs like line).

to construct a reduced load, by keeping the sequence of the most extreme forces during steering events. We further proposed a random load model by first, modeling the sequence of steering events, which is vehicle independent information, with a two states Markov chain. Then, assuming that the extreme forces occurring during the steering events, which is vehicle specific information, are independent random variables. An explicit formula for the expected fatigue damage was presented.

The proposed models were validated using measured data from a Volvo truck. The results show that all large rainflow cycles found in measured load were found in the reduced load. Hence, the fatigue damage of steering components can be predicted by reduced load. By simulations, we showed that the observed load spectrum did not significantly differ from the load spectra found in the simulated loads. It can be concluded that the proposed load model accurately describe the variability of the rainflow ranges for the considered measured load. Further, using the model the expected damage could be predicted.

Acknowledgements

We are thankful to Dr. Pär Johannesson for his helpful suggestions and valuable comments which improved the paper. 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.

Appendix A

Proof of Theorem 2

Consider the stationary time series X_i defined in Eq. (6). Let k be the number of maneuvers and random load $X = \{X_i : i = 0, \ldots, n\}$, $n = 2k$. We recall that $\mu_n^{osc}(u, v) = E[N_n^{osc}(u, v)]$ is the expected number of upcrossings of the interval $[u, v]$ found in X while the intensity of interval upcrossings

$$\mu^{osc}(u, v) = \lim_{n\to\infty} \frac{1}{n} \mu_n^{osc}(u, v).$$

We begin with a definition of $N_n^{osc}(u, v)$. Consider a load starting at time zero, i.e. infinite sequence $x = (x_0, x_1, \ldots)$ of real numbers. For fixed u, v, $u \le v$, $i \ge 0$ and $j > i + 1$ define the following sets:

$$\begin{aligned} A_i &= \{x_i < u\} \cap \{x_{i+1} > v\}, \\ A_{ij} &= \{x_i < u\} \cap \{x_j > v\} \cap \{u \le x_l \le v \text{ for all } l,\ i < l < j\}. \end{aligned} \tag{12}$$

Let $\mathbf{1}_A(x)$ be the indicator function of A, i.e. equal to one if $x \in A$ and zero otherwise. Now for fixed $u \le v$, $m \ge 2$ and $x \in R^{m+1}$ we will denote the number of upcrossings of interval $[u, v]$ found in x, i.e. $N_m^{osc}(u, v)$, by $N_m(x)$. Using the sets A_i and A_{ij}, defined in Eq. (12), one obtain that

$$N_m(x) = \sum_{i=0}^{m-1} \mathbf{1}_{A_i}(x) + \sum_{i=0}^{m-2} \sum_{j=i+2}^{m} \mathbf{1}_{A_{ij}}(x). \tag{13}$$

Now, we turn to evaluation of $\mu_n^{osc}(u, v) = E[N_n(X)]$ for $n = 2k$. The domain of μ_n^{osc} is divided into three regions; $u \le v < 0$, $0 < u \le v$ and $u \le 0 \le v$.

Region $u \le v < 0$: Since $X_i=0$ for all odd indices then $N_n(X)$ is equal to number of $X_{2i} < u < 0$, $0 \le i \le k - 1$. Consequently $\mu_n^{osc}(u, v) = k\pi_2 P(m_1 < u)$ and $\mu^{osc}(u, v) = \frac{1}{2}\pi_2 P(m_1 < u)$. (Recall that m_i, M_i are independent sequences of iid random variables.)

Region $0 < u \le v$: Similarly $N_n(X)$ is equal to number of $X_{2i} > v > 0$ and hence $\mu_n^{osc}(u, v) = k\pi_1 P(M_1 > v)$. Consequently $\mu^{osc}(u, v) = \frac{1}{2}\pi_1 P(M_1 > v)$.

Region $u \le 0 \le v$: Computation of $\mu_n^{osc}(u, v)$ and $\mu^{osc}(u, v)$ when $u \le 0 \le v$ is more complex. Let Y_i denote the sequence of extreme loads:

$$Y_i = \begin{cases} M_i & \text{if } Z_i = 1, \\ m_i & \text{if } Z_i = 2. \end{cases} \tag{14}$$

The number of crossings of intervals $[u, v]$, $u \le 0 \le v$ found in sequences X_i, $i = 0, \ldots, n$, and Y_i, $i = 0, \ldots, k$, are equal, i.e.

$$N_n(X) = N_k(Y), \quad Y_i = X_{2i}.$$

Now from the definition of $N_k(Y)$ in Eq. (13), it is easy to see that

$$\mu_n^{osc}(u,v)=\sum_{i=0}^{k-1}P(Y\in A_i)+\sum_{i=0}^{k-2}\sum_{j=i+2}^{k}P(Y\in A_{ij}).$$

Since Y_i is a stationary sequence hence $P(Y\in A_i)=P(Y\in A_0)$ and $P(Y\in A_{ij})=P(Y\in A_{0(j-i)})$ for any $j\geq i+2$. Consequently, with $P_1=P(Y\in A_0)$ and $P_l=P(Y\in A_{0l})$, $l=2,3,\ldots$,

$$\mu_n^{osc}(u,v)=kP_1+\sum_{i=0}^{k-2}\sum_{j=i+2}^{k}P_{j-i}=kP_1+\sum_{i=0}^{k-2}\sum_{l=2}^{k-i}P_l=\sum_{i=1}^{k}(k-i+1)P_i.$$

Hence the intensity $\mu^{osc}(u,v)$ is given by

$$\mu^{osc}(u,v)=\lim_{n\to\infty}\frac{1}{n}\mu_n^{osc}=\lim_{k\to\infty}\frac{1}{2}\sum_{i=1}^{k}(1-(i-1)/k)P_i=\frac{1}{2}\sum_{i=1}^{\infty}P_i,$$

by dominated convergence theorem ($\sum_{i=1}^{\infty}P_i\leq 1$). Next we will employ Markov property to evaluate $\mu^{osc}(u,v)$.

Let introduce the following sequence of events B_i, $i\geq 1$:

$$B_1=\{Y_1>v\}$$
$$B_i=\{Y_i>v \text{ and } u\leq Y_l\leq v \text{ for all } 1\leq l<i\},\quad i>1. \tag{15}$$

Using B_i the sum $\sum_{i=1}^{\infty}P_i$ can be written as follows:

$$\begin{aligned}\sum_{i=1}^{\infty}P_i&=P(Y\in A_0 \text{ and } Z_0=2)+\sum_{i=2}^{\infty}P(Y\in A_{0i}\text{ and } Z_0=2)\\&=\sum_{i=1}^{\infty}P(B_i|Z_0=2,Y_0<v)P(Y_0<v,Z_0=2)\\&=\pi_2P(m_0<u)\sum_{i=1}^{\infty}P(B_i|Z_0=2).\end{aligned} \tag{16}$$

Since probabilities p_j introduced in Section 3.2 are given by

$$p_j(u,v)=\sum_{i=1}^{\infty}P(B_i|Z_0=j),\quad j=1,2,$$

one has that $\sum_{i=1}^{\infty}P_i=\pi_2P(m<u)p_2(u,v)$. This finishes the proof of Eq. (8).

Finally we demonstrate that $p_2(u,v)$ is the solution of Eq. (7). Using Markov property one can evaluate p_j in the following way:

$$p_j(u,v)=P(Y_1>v|Z_0=j)+\sum_{l=1}^{2}\sum_{i=2}^{\infty}P(B_i|u\leq Y_1\leq v,Z_1=l,Z_0=j)$$

$$\cdot P(u\leq Y_1\leq v,Z_1=l|Z_0=j)=P(Y_1>v|Z_0=j)+\sum_{l=1}^{2}\sum_{i=2}^{\infty}P(B_i|Z_1=l)$$

$$P(u\leq Y_1\leq v|Z_1=l)p_{jl}=P(Y_1>v|Z_0=j)+\sum_{l=1}^{2}P(u\leq Y_1\leq v|Z_1=l)$$

$$p_{jl}\sum_{i=1}^{\infty}P(B_i|Z_0=l)=P(Y_1>v|Z_0=j)+\sum_{l=1}^{2}P(u\leq Y_1\leq v|Z_1=l)$$

$$p_{jl}P_l(u,v).$$

This completes the proof.

References

[1] H. Berndt, K. Dietmayer, Driver intention inference with vehicle onboard sensors, in: IEEE International Conference on Vehicular Electronics and Safety (ICVES), Pune, 11–12 November 2009, pp. 102–107.
[2] A. Beste, K. Dressler, H. Kötzle, W. Krüger, B. Maier, J. Petersen, Multiaxial rainflow—a consequent continuation of Professor Tatsuo Endo's work, in: Y. Murakami (Ed.), The Rainflow Method in Fatigue, Butterworth-Heinemann, 1992, pp. 31–40.
[3] Bishop, Sherratt, A theoretical solution for estimation of rainflow ranges from power spectral density data, Fatigue Fract. Eng. Mater. Struct. 13 (1990) 311–326.
[4] K. Bogsjö, K. Podgórski, I. Rychlik, Models for road surface roughness, Veh. Syst. Dyn. 50 (2012) 725–747.
[5] P.A. Brodtkorb, P. Johannesson, G. Lindgren, I. Rychlik, J. Rydén, E. Sjö, WAFO—a Matlab toolbox for analysis of random waves and loads, in: Proceedings of the 10th International Offshore and Polar Engineering Conference, Seattle vol. III, 2000, pp. 343–350.
[6] O. Cappé, E. Moulines, T. Rydén, Inference in Hidden Markov Models, Springer, New York, 2005.
[7] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via EM algorithm, J. R. Stat. Soc. Ser. B Methodol. 39 (1) (1977) 1–38.
[8] M. Frendahl, I. Rychlik, Rainflow analysis—Markov method, Int. J. Fatigue 15 (1993) 265–272.
[9] G. Lindgren, K.B. Broberg, Cycle range distributions for Gaussian processes—exact and approximate results, Extremes 7 (2004) 69–89.
[10] P. Johannesson, Rainflow cycles for switching processes with Markov structure, Probab. Eng. Inf. Sci. 12 (1998) 143–175.
[11] P. Johannesson, Rainflow analysis of switching Markov loads (Ph.D. thesis), Lund Institute of Technology, 1999.
[12] P. Johannesson, M. Speckert (Eds.). Guide to Load Analysis for Durability in Vehicle Engineering, Wiley, Chichester, 2013.
[13] M. Karlsson, Load modelling for fatigue assessment of vehicles—a statistical approach (Ph.D. thesis), Chalmers University of Technology, Sweden, 2007.
[14] S. Kotz, T. Kozubowski, K. Podgorski, The Laplace distribution and generalizations: a revisit with applications to communications economics, engineering, and finance, Birkhäuser Boston, Inc., Boston, MA, 2001.
[15] S. Krenk, H. Gluver, A Markov matrix for fatigue load simulation and rainflow range evaluation, Struct. Saf. 6 (1989) 247–258.
[16] M. Kvanström, K. Podgórski, I. Rychlik, Laplace moving average model for multi-axial responses in fatigue analysis of a cultivator, Probab. Eng. Mech. 34 (2013) 12–25.
[17] R. Maghsood, A statistical approach for detecting driving events and evaluating their fatigue damage (Lic. thesis), Chalmers University of Technology, 2014.
[18] R. Maghsood, P. Johannesson, Detection of the curves based on lateral acceleration using hidden Markov models, Proc. Eng. 66 (2013) 425–434.
[19] M. Matsuishi, T. Endo, Fatigue of Metals Subjected to Varying Stress, Japan Society of Mechanical Engineers , 1968 (in Japanese).
[20] M.A. Miner, Cumulative damage in fatigue, J. Appl. Mech. 12 (1945) A159–A164.
[21] D. Mitrović, Learning driving patterns to support navigation (Ph.D. thesis), University of Canterbury, New Zealand, 2004.
[22] D. Mitrović, Reliable method for driving events recognition, IEEE Trans. Intell. Transp. Syst. 6 (2) (2005) 198–205.
[23] A. Palmgren, Die Lebensdauer von Kugellagern, Z. Vereins Dtsch. Ing. 68 (1924) 339–341 (in German).
[24] I. Rychlik, A new definition of the rainflow cycle counting method, Int. J. Fatigue 9 (1987) 119–121.
[25] I. Rychlik, Rain flow cycle distribution for ergodic load processes, SIAM J. Appl. Math. 48 (1988) 662–679.
[26] I. Rychlik, Note on cycle counts in irregular loads, Fatigue Fract. Eng. Mater. Struct. 16 (1993) 377–390.
[27] I. Rychlik, Note on modelling of fatigue damage rates for non-Gaussian stresses, Fatigue Fract. Eng. Mater. Struct. 36 (2013) 750–759.
[28] I. Rychlik, G. Lindgren, Y.K. Lin, Markov based correlations of damages in Gaussian and non-Gaussian loads, Probab. Eng. Mech. 10 (1995) 103–115.
[29] A.J. Viterbi, Error bounds for convolutional codes and an asymptotically optimal decoding algorithm, IEEE Trans. Inf. Theory IT-13 (2) (1967) 260–269.
[30] WAFO Group, WAFO—A Matlab Toolbox for Analysis of Random Waves and Loads, Tutorial for WAFO 2.5, Mathematical Statistics, Lund University, 2011.
[31] WAFO Group, WAFO—A Matlab Toolbox for Analysis of Random Waves and Loads Version 2.5 07-Feb-2011, Mathematical Statistics, Lund University, 2011. Web: ⟨http://www.maths.lth.se/matstat/wafo/⟩ (accessed 24 January 2014).

Paper III

Load Description and Damage Evaluation using Vehicle Independent Driving Events

R. Maghsood, I. Rychlik and P. Johannesson

Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 101 (2015) 268 – 276

Procedia Engineering

www.elsevier.com/locate/procedia

3rd International Conference on Material and Component Performance under Variable Amplitude Loading, VAL2015

Load Description and Damage Evaluation using Vehicle Independent Driving Events

Roza Maghsood[a,*], Igor Rychlik[a], Pär Johannesson[b]

[a]*Chalmers University of Technology , SE-412 96 Gothenburg, Sweden*
[b]*SP Technical Research Institute of Sweden, SE-400 22 Gothenburg, Sweden*

Abstract

We consider the loads that are related to steering events, and focus on the events that cause high forces on steering components. The load is simplified by keeping the extreme force value for each driving event. We define a simplified stochastic model for the load by modeling the extreme value for each driving event by a random variable. We give formulas to compute the theoretical load spectrum and the expected fatigue damage caused by the driving events. Further, in a sensitivity study we investigate how much the expected damage depends on the variability of parameters of the proposed model.

Peer-review under responsibility of the Czech Society for Mechanics

Keywords: Fatigue damage index; hidden Markov models; Markov chain; rainflow cycles; vehicle independent load models; steering events.

1. Introduction

In vehicle engineering, durability is an important aspect of designing a vehicle with high quality in its components. Therefore, considering the service loading conditions is necessary. In addition, in fatigue design the loads need to be assessed. By describing the load environment, the customer usage and the vehicle dynamics one can define the load conditions [1].

* Corresponding author. Tel.: +46 31 772 53 79.
E-mail address: rozam@chalmers.se

Peer-review under responsibility of the Czech Society for Mechanics
doi:10.1016/j.proeng.2015.02.026

Nomenclature

λ	hidden Markov model parameters	d_β	damage intensity
$D_\beta(x)$	damage index	$E\left[D_\beta(X)\right]$	expected damage index
α , β	material constants for Palmgren–Miner's rule	$N^{rfc}(u,v)$	rainflow counting distribution
u , v	lower and upper ranges for cycle range	$\mu^{osc}(u,v)$	intensity of interval crossings

For vehicle companies, it is important to characterize the way that the trucks have been used. They want to describe the usage of the trucks in a way that it is independent of the vehicle. The loads will be different for different usage of trucks and for different driver's behavior. A driver can affect the load by changing the speed, braking or adapting to the curves. These behaviors can be characterized as driving events and can be assessed using measurements obtained from specially equipped vehicles on a test track. Measuring the load on each truck is expensive. However, they want to measure and identify activities of the driver and specify the relevant events occurring on the road. To identify the events we need to use the information available for all vehicles by means of CAN (Controller Area Network) bus data. If we define the events such as static steering by using the information from CAN data, we can detect the amount of events that are occurring in customer vehicles. Then, it is possible to calculate the forces generated from the same kind of events by repeating the loads under well-defined conditions on a proving ground. By using the force signal we can clarify which occasions will generate high forces.

We have proposed a stochastic model of loads related to the steering events such as curves and maneuvers, which cause large forces acting on steering components. An explicit formula for calculating the expected fatigue damage based on identified driving events is given, see also Maghsood and Rychlik [2]. The expected damage depends on the frequencies of driving events and the expected value of the extreme force during an event. The model consists of two parts; description of the sequence of steering events and the model for the extreme loads occurring during the events. The sequence of steering events is modeled by means of a Markov chain. This is a vehicle independent part of the load. For simplicity, the extreme forces during the events are assumed to be statistically independent. Their distributions may depend on the type of steering event, e.g. (left, right) cornering, slow maneuver to the right or to the left etc. The parameters of the distributions are vehicle dependent and need to be estimated using dedicated measurement campaigns or test track measurements. In the examples in Sections 5 and 6, the Rayleigh distribution will be used to describe the variability of extreme forces. Further, the uncertainty in fatigue damage due to model parameters will be discussed.

The paper is organized as follows. In Section 2 hidden Markov models (HMMs) based algorithm to detect the steering events is reviewed. The proposed model for loads and means to calculate the expected damage are described in Sections 3 and 4. Examples and their results for measured data are shown in Section 5. In Section 6 the sensitivity analyses are investigated. Conclusions are presented in Section 7.

2. Detection of the steering events

Hidden Markov models (HMMs) have been proposed for detection of steering events such as curves and maneuvering using on-board logging signals available on trucks, such as lateral acceleration, vehicle speed and steering wheel angle. The idea is to consider the current driving event as the hidden state and set up the model based on them, see Maghsood and Johannesson [3, 4].

We have 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. The parameters must be estimated to characterize the model, see Rabiner [5] for more details.

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 the examples, the training set contains all history about the curves such as the start and stop points of them. Fig. 1 shows a lateral acceleration signal and the corresponding identified hidden states process.

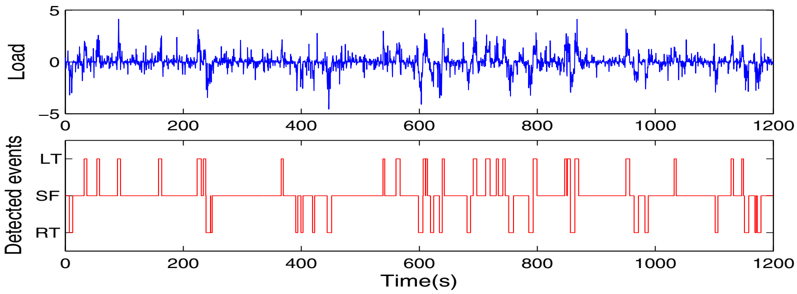


Fig. 1. Lateral acceleration signal and the corresponding detected events.

3. Random model of lateral loads based on steering events

Modeling of the external loads is an important aspect in durability studies of vehicle components. The approach taken here is to approximate the load by a vehicle independent sequence of steering events, here representing Left and Right steering (SL, SR) or Left and Right turns (LT, RT). In both cases the two events are separated by a section when wheels have approximately zero turning angle, which is called Straight forward (SF). Thus, a reduced load can be defined by keeping the extreme value for each left and right event and set zero for each straight forward event. The most extreme value of the load will be modeled by a random variable Y_i . First the variability of the sequence of steering events is modeled by a Markov chain Z_i having two states "1" and "2", then the values of extreme forces during events will be modeled. The Markov chain is defined by a transition matrix $P = (p_{ij}), i, j = 1,2$, where p_{ij} denotes the transition probabilities between the states.

Let $M_i, i = 0,1,2,...$ be a sequence of independent and identically distributed (iid) positive random variables while $m_i, i = 0,1,2,...$ denotes the negative random variables. Assume that the three sequences $\{Z_i\}_{i=0}^{\infty}, \{M_i\}_{i=0}^{\infty}$ and $\{m_i\}_{i=0}^{\infty}$ are independent. The process Z_i is vehicle independent while M_i and m_i depend on the vehicle, driver etc. The sequence of extreme loads $Y_i, i = 0,1,2,...$, is defined by:

$$Y_i = \begin{cases} M_i, & if \quad Z_i = 1, \\ m_i, & if \quad Z_i = 2. \end{cases} \tag{1}$$

Finally, we can define the random load $X_i, i = 0,1,2,...$ by adding zeros between Y_i and Y_{i+1} for each straight forward event:

$$X_i = \begin{cases} 0, & if \quad i \quad is \quad odd, \\ Y_{i/2}, & otherwise \end{cases} \tag{2}$$

4. Fatigue damage index

The aim is to compute the expected damage based on the detected driving events. To evaluate the model, we will compare the estimated damage index from the measured forces using rainflow method with the expected damage from the proposed load model. To calculate the damage, we have used forces which are measured from specially equipped vehicles on a test track. First we will review some models and methods on fatigue damage.

Assume that the measured load x is given in form of time series $x_i, i = 0,1,2,...,n$. The risk for fatigue failure in the material is often measured by means of a damage index which can be computed by Palmgren-Miner rule [6, 7], viz.

$$D_\beta(x) = \sum_{i=1}^{N} \frac{1}{N_i} = \alpha \sum_{i=1}^{N} h_i^\beta \tag{3}$$

where N_i is the number of cycles having ranges h_i to failure estimated in constant amplitude tests and presented in form of S-N curve.. The parameter α is the fatigue strength of the material and β is the damage exponent.

The variability of the load is modelled by means of random processes. Therefore, the measured load x is one of many possible realizations of the process. For the random loads, the rainflow ranges become random variables and the damage index is a random quantity too. The variability of the rainflow cycles can be described using a cumulative histogram $N^{rfc}(u,v)$ which is called the rainflow counting distribution. The rainflow counting distribution $N^{rfc}(u,v)$ is equal to the number of times that the load $x_i, i = 0,1,2,...,n$, crosses an interval $[u,v]$ in upward direction, denoted by $N_n^{osc}(u,v)$. The equality between the rainflow counting distribution and the interval crossing was shown independently in [8] and [9].

The damage intensity can be used to measure the severity of the random load and it can be computed using the intensity of interval crossings:

$$\mu^{osc}(u,v) = \lim_{n\to\infty} \frac{E[N_n^{osc}(u,v)]}{n}, \tag{4}$$

then the damage intensity is

$$d_\beta = \beta(\beta-1)\int_{-\infty}^{\infty}\int_{-\infty}^{v}(v-u)^{\beta-2}\mu^{osc}(u,v)\,dudv. \tag{5}$$

The main result is an explicit formula for $\mu^{osc}(u,v)$ based on the random load X_i :

$$\mu^{osc}(u,v) = \frac{1}{2}\begin{cases} \pi_2 P(m_1 < u), & u < v < 0 \\ \pi_2 P(m_1 < u) p_2(u,v), & u \le 0 \le v \\ \pi_1 P(M_1 > v), & 0 < u < v \end{cases} \tag{6}$$

where $p_2(u,v)$ is the solution to the equation system

$$\begin{aligned} p_1(u,v) &= p_{11}P(M_1 > v) + P(M_1 \le v)p_{11}p_1(u,v) + P(m_1 \ge u)p_{12}p_2(u,v), \\ p_2(u,v) &= p_{21}P(M_1 > v) + P(M_1 \le v)p_{21}p_1(u,v) + P(m_1 \ge u)p_{22}p_2(u,v). \end{aligned} \tag{7}$$

see [6] for more details and prof of formula (6).

5. Example

The results are presented for maneuvering events. The curves were also studied in [2] but the results will not be presented here. The maneuvering events, i.e. driving in or out of a parking lot, standing still but turning steering wheel,

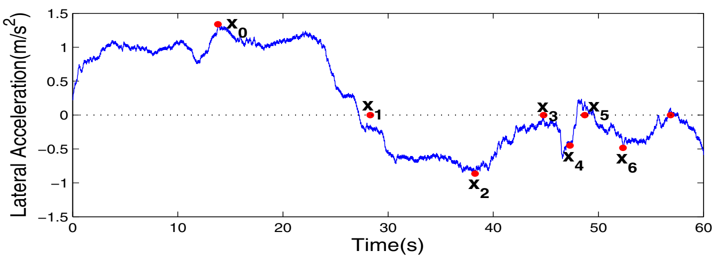
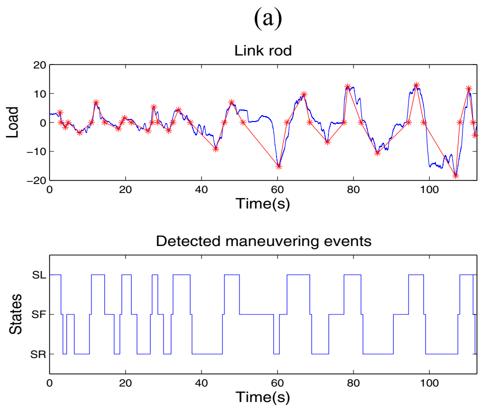


Fig. 2. Reduced load represented by dots compared with observed load, lateral acceleration, represented by the irregular solid line.

are considered as the events which will happen in speed less than about 10 km/h. Here, three maneuvering events are considered; Steering Left (SL), Steering Right (SR) and Straight Forward (SF). The measured loads are denoted by x^{obs}. First, the steering events were detected using HMM algorithm, then the extreme loads during events were found. We assume that each event follows by a driving straight section. The signal consisting of the extreme loads during steering events and zeros for section when vehicle is driving straight will be denoted by $x = (x_0, x_1, \ldots x_n)$ and called the *reduced load*. In Fig. 2 part of measured load x^{obs} (lateral acceleration) is shown as the solid line while the reduced load x by dots.

The link rod force is used as the load and it is shown in the top plots of Fig. 3a. The extreme forces are negative, positive and zero in the three states SR, SL and SF, respectively. In the figure stars are the extreme rod forces, occurring during maneuvers, constituting the reduced load. In the lower plot of Fig. 3a, the detected time periods with 21 detected maneuvering events are shown.



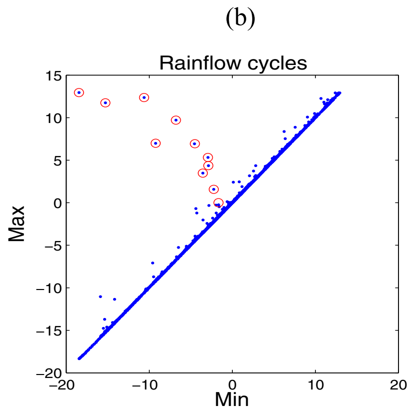


Fig. 3. (a) Top: solid irregular line is the measured link rod force while stars represent the reduced load. Bottom: Detected maneuvers. (b) Dots - the rainflow cycles found in the measured link rod force. Circles - the rainflow cycles counted in the reduced load.

The rainflow cycles have been found both in the load and in the reduced load and compared in Fig. 3b. The rainflow cycles found in the measured link rod force are marked as dots. One can see that there are few large cycles and many very small ones. The rainflow cycles found in the reduced load are presented as circles. As can be seen in Fig. 3b, all large cycles found in the link rod force are also found in the reduced load and hence one can expect that the damage index computed for the measured load and the reduced load should be very close.

The estimated transition matrix according to the detected maneuvers is

$$P = \begin{pmatrix} 0 & 1 \\ 0.9 & 0.1 \end{pmatrix}.$$

The Rayleigh distributions have been fitted to positive and negative values, respectively. The estimates of the parameters of Rayleigh distributions were very close. The difference between the parameter values were not significant hence the average value (6.1) of the parameters have been used.

Table 1 shows a comparison of the damage indexes $D_\beta(x^{obs})$ computed for measured load, $D_\beta(x)$ for the reduced load and the expected damage index $E[D_\beta(X)]$ for the random model of the reduced load. Damage indices $D_\beta(x^{obs})$ and $D_\beta(x)$ are given in columns 2 and 3. As expected these are almost identical. We conclude that the reduced load models well the variability of the measured load. Further, the expected damage of the model is quite close to the measured one.

Table 1. Comparison of damage indices $D_\beta(x^{obs})$ computed for the measured load, $D_\beta(x)$ for the reduced load and the expected damage index $E\lfloor D_\beta(X)\rfloor$.

Damage	$D_\beta(x^{obs})$	$D_\beta(x)$	$E\lfloor D_\beta(X)\rfloor$
$\beta = 3$	9.39×10^{3}	9.34×10^{3}	8.35×10^{3}
$\beta = 5$	1.70×10^{6}	1.70×10^{6}	1.47×10^{6}

In Fig. 4a, the load spectra estimated from the measured link rod force and the reduced load are compared with the theoretical load spectrum. As can be seen in Fig. 4b, where the load spectra for 10 simulated loads are compared with the theoretical load spectrum and the load spectrum of the reduced load, the differences between the measured spectrum and the expected one does not seem to be significant.

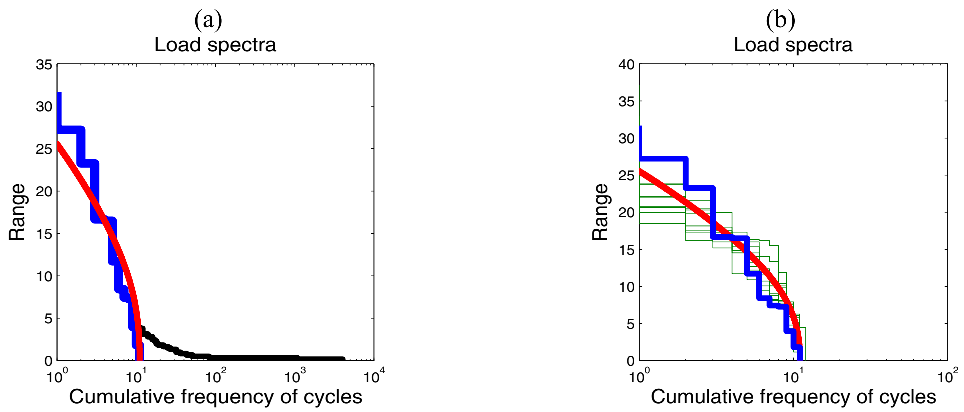


Fig. 4. (a) The regular solid line is the theoretical load spectrum. The stairs like functions are the load spectra found in measured link rod force and the reduced load. (b) Load spectra for 10 simulated loads compared with the theoretical load spectrum and the load spectrum of the reduced load (the thick stairs like line).

6. Sensitivity analysis of the damage index

As it was mentioned before, the sequence of steering events is modeled by a Markov chain with transition matrix P. This sequence is a vehicle independent part of the load. The extreme forces during the events are assumed to be statistically independent, but their distributions depend on the type of steering event. The parameters of the distributions are vehicle dependent and need to be estimated using dedicated measurement campaigns or test track measurements. In the examples, Rayleigh distributions have been fitted to positive and negative values. Now suppose that both distributions have the same parameter σ, viz.

$$P(M_1 > v) = e^{-\frac{1}{2}(\frac{v}{\sigma})^2}, v \geq 0 \qquad P(m_1 < u) = e^{-\frac{1}{2}(\frac{u}{\sigma})^2}, u \leq 0 \tag{8}$$

then the load can be written as a scaled standard load, $X = \sigma\hat{X}$, where $\hat{X}$ is a reduced load with standard Rayleigh random variables, $P(R > r) = e^{-r^2/2}$. Therefore, the oscillation intensity can be written as a scaled one, namely

$$\mu^{osc}(u,v) = \hat{\mu}^{osc}(\frac{u}{\sigma}, \frac{v}{\sigma}). \tag{9}$$

Further, the damage index can be calculated as $d_\beta = \sigma^\beta \cdot \hat{d}_\beta$, where $\hat{d}_\beta$ is the expected damage computed by the standard Rayleigh distribution. This means that the expected damage index is a factor of the parameter σ to the power β.

Here, we will consider two types of uncertainties. The variability of the load environment will manifest in the transition matrix P and the vehicle dependent variability in the parameter σ of the Rayleigh distribution. In the following subsections we will study how much the expected damage will vary because of variability of matrix P and parameter σ. We will also investigate the statistical uncertainty of the estimation of σ. In fatigue reliability evaluation using the load-strength concept often the log-normal distribution is used, see [1, Chapter 7]. Therefore, the uncertainty in damage will be measured in terms of the standard deviation of the logarithmic damage, which corresponds to the relative uncertainty in damage (or fatigue life).

6.1. Variability of the transition matrix P

To examine how much the expected damage will depend on the transition matrix P, three different Markov chains have been used to model the sequence of driving events.

- First, assume that we always go from left to left. This would be the case with the smallest possible damage, since all minima are equal to zero. The expected damage is $E[D] = nE[R^{\beta}]$, where n denotes the number of turns and R represents the standard Rayleigh random variable for the maximum force.
- Second, consider that the events change each time. In this case, $p_{12} = p_{21} = 1$, and we will get the maximum damage for this type of Markov chain. The oscillation intensity $\mu^{osc}(u,v)$ given in Eq. (6) can be simplified to

$$\mu^{osc}(u,v) = \frac{1}{4}\begin{cases} P(m_1 < u), & u < v < 0 \\ \dfrac{P(M_1 > v)P(m_1 < u)}{1 - P(M_1 \le v)P(m_1 \ge u)}, & u \le 0 \le v \\ P(M_1 > v), & 0 < u < v \end{cases} \tag{10}$$

- Finally, assume that left and right turns occur independently of the past with probabilities $p_{ij} = 0.5$, and Eq. (6) simplifies to (see also [10])

$$\mu^{osc}(u,v) = \frac{1}{4}\begin{cases} P(m_1 < u), & u < v < 0 \\ \dfrac{P(M_1 > v)P(m_1 < u)}{P(M_1 > v) + P(m_1 < u)}, & u \le 0 \le v \\ P(M_1 > v), & 0 < u < v \end{cases} \tag{11}$$

The expected damage values for the three different cases have been summarized in Table 2. Here, we have considered standard Rayleigh distributions for negative and positive values and the number of events is $n = 100$.

Table 2. Expected damage calculated from the three different Markov chains.

Damage	Minimum	Independent	Maximum
$\beta = 3$	0.05×10^3	0.14×10^3	0.17×10^3
$\beta = 5$	0.06×10^3	0.72×10^3	0.80×10^3

The two extreme cases will be used to calculate the uncertainty in damage by assuming a uniform distribution between the minimum and maximum values, viz. for $\beta = 3$

$$\tau_P = \frac{\ln d_{\max} - \ln d_{\min}}{\sqrt{12}} = \frac{\ln 0.17 - \ln 0.05}{\sqrt{12}} = 0.35 \tag{12}$$

which can be interpreted as corresponding to 35% relative uncertainty in damage, as the natural logarithm is used.

6.2. Statistical uncertainty of parameter σ

Suppose that we estimate the parameter σ based on n observations, then we may ask how much the estimation uncertainty of parameter σ impacts the damage. The estimate of parameter σ for a Rayleigh distribution is $\hat{\sigma} = \sqrt{2/\pi}\,\overline{X}$ and its distribution is approximately normal $N(\sigma, \frac{4-\pi}{n\pi}\sigma^2)$. Thus, the uncertainty in damage can be approximated using Gauss' approximation formula, viz.

$$\tau_{\sigma,stat} = \mathrm{Var}\left[\ln \hat{\sigma}^{\beta}\right] = \beta \cdot \mathrm{Var}\left[\ln \hat{\sigma}\right] \approx \beta \cdot \sqrt{\frac{4-\pi}{n\pi}} = 0.29. \tag{13}$$

with an example for a short signal with $n = 30$ maneuvering events and $\beta = 3$, corresponding to a typical length of the measurements.

6.3. Variability of parameter σ

For different measurements of maneuvering events, we have found different estimates of parameter σ, say $\sigma_1, \sigma_2, ..., \sigma_l$ for l different measurements. The uncertainty in damage due to the variability in the estimated σ is computed as the sample standard deviation, viz.

$$\tau_{\sigma} = \mathrm{std}\left(\ln \sigma^{\beta}\right) = \beta \cdot \mathrm{std}\left(\ln \sigma\right) = 3 \cdot 0.18 = 0.56 \tag{14}$$

for an example with $\beta = 3$ and estimated σ-values 6.15, 6.05, 9.40, 8.96, 7.76, 6.37, 6.72. However, the uncertainty τ_{σ} includes both the variability and the statistical uncertainty of σ. Thus, the pure variability can be estimated as $\tau_{\sigma,\mathrm{var}} = \sqrt{\tau_{\sigma}^2 - \tau_{\sigma,stat}^2} = 0.46$.

7. Conclusion

A reduced load, i.e. a sequence of the most extreme forces during steering events, was introduced. A random load modeling the variability of the reduced load was proposed. The sequence of steering events, which is vehicle independent information, was modeled using a two states Markov chain. The extreme forces occurring during the steering events were modeled by means of independent Rayleigh distributed variables. For the model, an explicit formula for the expected fatigue damage was presented. The proposed random model depends only on four parameters which could be used to classify and compare the severity of driving environments.

The results were validated using measured data. The slow speed maneuvering events were detected. All large rainflow cycles found in measured load were also counted in the reduced load. Hence the reduced load can be used to predict fatigue damage of steering components. The observed load spectrum did not significantly differ from load spectra found in the simulated loads. We conclude that the proposed random load accurately describe the variability of the rainflow ranges for the considered measured loads.

A sensitivity study was conducted to see how much the expected damage depends on the parameters of the proposed model.

Acknowledgements

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. We gratefully acknowledge the financial support from VINNOVA.

References

[1] P. Johannesson and M. Speckert, editors. *Guide to Load Analysis for Durability in Vehicle Engineering*, Wiley:Chichester, 2013.

[2] R. Maghsood and I. Rychlik, Estimation of fatigue damage of steering components using vehicle independent load model, Submitted to Probabilistic Engineering Mechanics, August 2014.

[3] R. Maghsood and P. Johannesson, Detection of the curves based on lateral acceleration using hidden Markov models, *Procedia Engineering*, 66:425-434, 2013.

[4] R. Maghsood and P. Johannesson. Detection of the steering events based on vehicle logging data using hidden Markov models. Submitted to International Journal of Vehicle Design, April 2014.

[5] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77(2): 257 - 286, 1989.

[6] A. Palmgren, Die Lebensdauer von Kugellagern. *Zeitschrift des Vereins Deutscher Ingenieure*, 68:339-341, 1924. *In German*.

[7] M. A. Miner. Cumulative damage in fatigue. *Journal of Applied Mechanics*, 12:A159-A164, 1945.

[8] A. Beste, K. Dressler, H. Kötzle, W. Krüger, B. Maier, and J. Petersen. Multiaxial rainflow - a consequent continuation of Professor Tatsuo Endo's work. In Y. Murakami, editor, *The Rainflow Method in Fatigue*, pages 31- 40. Butterworth-Heinemann, 1992.

[9] I. Rychlik. Note on cycle counts in irregular loads. *Fatigue & Fracture of Engineering Materials & Structures*, 16:377-390, 1993.

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

Paper IV

Detection of Steering Events using Hidden Markov Models with Multivariate Observations

R. Maghsood, P. Johannesson and J. Wallin

Detection of steering events using hidden Markov models with multivariate observations

R. Maghsood*[a], P. Johannesson[b] and J. Wallin[a]
[a]Chalmers University of Technology
[b]SP Technical Research Institute of Sweden
*Corresponding author

Abstract

In this article we propose a method to identify steering events, such as curves and maneuvers for vehicles. We use a hidden Markov model with multidimensional observations, to estimate the number of events. Three signals, lateral acceleration, steering angle speed and vehicle speed are used as observations. We demonstrate that hidden Markov models with a combination of continuous and discrete distributions for observations can be used to detect steering events. Further, the expected number of events is estimated using the transition matrix of hidden states. The results from both measured and simulated data show that the method works well and accurately estimates the number of steering events.

Keywords: Hidden Markov models (HMMs); Laplace distribution; discrete distribution; EM algorithm; steering events.

1 Introduction

For vehicle companies, the durability assessment of vehicle components is an important aspect of the design process to produce a vehicle with high quality of its components. Therefore, describing the service loading independent of vehicle properties is appreciated at the design stage of the components. In addition, for fatigue damage the loads need to be assessed. The load environment, the customer usage and the vehicle dynamics affect the load distribution, Johannesson and Speckert (2013).

One of the main sources of variation in the loads is the driver's behavior. A driver can affect the load by changing the speed, braking or adapting to the curves. These behaviors are characterized as driving events. In general, the customer behavior is unknown and needs to be estimated. To identify the events we want to use the information available for all vehicles by means of CAN (Controller Area Network) bus data. Since the driving events are not recorded in CAN data, we model them using a hidden Markov model, Cappé et al. (2005), where the states are the driving events and the observations are CAN signals.

There are different methods for modeling driver actions and identifying driving events. For instance, Nilsson et al. (2014) have developed an on-line cycle detection algorithm to extract detailed information from customers' vehicles during operation, by using only production sensors. Karlsson (2007) has described customer usage by classifying the type of roads. This is important since different aspects of the customer usage are relevant in order to determine the fatigue damage in service. Such aspects are the type of roads, transport mission, driver's behavior and different kind of maneuvers they perform.

Hidden Markov models are probabilistic models often used in signal processing for detection of patterns or events in a signal. The HMMs are the reliable and robust methods for event recognition. The idea of using HMMs to identify driving events is not new and it has been used in many applications. Mitrović (2004, 2005) and Berndt and Dietmayer (2009) have constructed one HMM for each type of driving event such as left and right curves, left, right and straight on in roundabouts. In our study, we have used a single HMM for describing all driving events.

In Maghsood and Johannesson (2013, 2016), a discrete HMM was used to detect steering events, either curves or maneuvering, based on vehicle logging data. The discretization of the observation process was obtained by using predefined thresholds, and then the probability distribution of observation symbols are considered in each hidden state. Two separate HMMs were constructed, one for detection of curves based on lateral acceleration, and another one for detection of maneuvers (steering activities at low speed) based on the steering angle speed.

It is also possible to use a continuous HMM for detecting steering events, as described in Maghsood et al. (2015). Here, also two separate HMMs for detection of curves and maneuvers are needed. However, there are two main advantages of using continuous HMM over a discrete version. Firstly, there is no need to choose the thresholding levels. Secondly, the continuous model can easily be extended to incorporate multivariate sources of information, which is not as easy for the discrete threshold approach.

In this study, we want to investigate the usefulness of HMMs with multidimensional observations to identify both the curves and the maneuvers in a single HMM. We use an HMM with a combination of continuous Laplace distributions and discrete distributions for observations. The multivariate observation consists of the lateral acceleration, the steering angle speed and the discretized vehicle speed, all calculated from CAN data. Both simulated and measured signals are used as examples to identify the steering events and validate the model.

The paper is organized as follows: In the second section, the HMM and the estimation method are presented. In the third section, the HMMs for steering events are introduced while the data is described in Section four. The examples are presented in Section five. The final section contains the conclusions of the paper.

2 Model description and estimation

We investigate the usefulness of HMMs for finding different driving events. Using a combination of continuous and discrete distributions for multidimensional observations in HMM is of interest. We start with a short description of HMMs and the multivariate distribution of observations. Then, we define the generalized asymmetric Laplace distribution (GAL) and the discrete distribution used to detect the sequence of steering events. Further, a method for estimating HMM parameters is presented in Section 2.3, while estimating the number of events is given in Section 2.4.

2.1 Hidden Markov models

Hidden Markov model is a bivariate Markov process $\{Z_t, Y_t\}_{t=0}^{\infty}$ where the underlying process Z_t is an unobservable Markov chain and is observed only through the observation sequence Y_t, see Cappé et al. (2005). The observation Y_t given Z_t is a sequence of independent random variables and the conditional distribution of Y_t depends only on Z_t.

In this article, the sequence of hidden states Z_t takes values on a discrete space $\{1, 2, \ldots, N\}$. The HMM is characterized by two sets of parameters. The first set is the transition matrix $\boldsymbol{Q} = (q(i, j))$ of Markov chain Z_t, where the transition probabilities $q(i, j)$ are given by:

$$q(i, j) = P(Z_{t+1} = j | Z_t = i),\ i, j = 1, 2, ..., N. \tag{1}$$

The second set is the parameters, $\boldsymbol{\theta}$, of the conditional distribution of Y_t given Z_t:

$$g_{\boldsymbol{\theta}}(i, y_t) = f_{Y_t}(y_t | Z_t = i; \boldsymbol{\theta}),\ i = 1, 2, ..., N. \tag{2}$$

The observation sequence Y_t can be a univariate or multivariate variable.

In an HMM, the state where the hidden process will start is modeled by the initial state probabilities $\boldsymbol{\pi} = (\pi_i)$, where π_i is denoted by:

$$\pi_i = P(Z_0 = i),\ i = 1, 2, ..., N \tag{3}$$

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

2.2 The distribution of observations

In this article, we propose using a combination of continuous and discrete distributions for observations. To estimate the parameters we use the maximum likelihood method. Therefore, we need to find the conditional distribution of $\boldsymbol{Y}_t$ given Z_t. Suppose that the observation $\boldsymbol{Y}_t = (Y_{t,1}, ..., Y_{t,d})$ is a multivariate time series with $d = d_1 + d_2$ dimensions. The first d_1 random variables $(Y_{t,1}, ..., Y_{t,d_1})$ are continuous with their observed values $(y_{t,1}, ..., y_{t,d_1})$, and have the joint probability density function $f_{Y_{t,1},...,Y_{t,d_1}}(y_{t,1}, ..., y_{t,d_1} | Z_t = i; \boldsymbol{\theta}_1)$ within each state. We assume that the last d_2 observations $(Y_{t,d_1+1}, ..., Y_{t,d_2})$ are discrete variables and have the joint probability mass function $P(Y_{t,d_1+1} = y_{t,d_1+1}, ..., Y_{t,d_2} = y_{t,d_2}|Z_t = i, \boldsymbol{\theta}_2)$ in each state.

For the sake of simplicity we assume that the continuous and discrete variables are conditionally independent given the hidden state. Then, the frequency function of $\boldsymbol{Y}_t$ given Z_t for a set of parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ is as follows:

$$\begin{aligned} g_{\boldsymbol{\theta}}(i, \boldsymbol{y}_t) &= f_{Y_{t,1},...,Y_{t,d_1}}(y_{t,1}, ..., y_{t,d_1}|Z_t = i; \boldsymbol{\theta}_1) \\ &\qquad P(Y_{t,d_1+1} = y_{t,d_1+1}, ..., Y_{t,d_2} = y_{t,d_2}|Z_t = i; \boldsymbol{\theta}_2) \end{aligned} \quad (4)$$

where $\boldsymbol{y}_t = (y_{t,1}, ..., y_{t,d})$ is the observed value of $\boldsymbol{Y}_t$ and $i = 1, 2, ..., N$.

Note that any continuous or discrete distributions can be used in Eq. (4). In our case study, on-board logging signals are used as observations $\boldsymbol{Y}_t$ and some of them are discretized into the predefined levels. We model the continuous observations by Laplace distributions and use a discrete distribution for discretized signals.

2.2.1 Multivariate generalized Laplace distribution

The multivariate generalized asymmetric Laplace distribution (GAL) for a d-dimensional random vector $\boldsymbol{Y}$ is denoted by $GAL(\boldsymbol{\delta}, \boldsymbol{\mu}, \nu, \boldsymbol{\Sigma})$, where $\boldsymbol{\delta}$ is the location vector, $\boldsymbol{\mu}$ is the shift vector, $\nu > 0$ is the shape parameter and $\boldsymbol{\Sigma}$ is the scaling matrix. The probability density function (pdf) of a $GAL(\boldsymbol{\delta}, \boldsymbol{\mu}, \nu, \boldsymbol{\Sigma})$ distribution is

$$\begin{aligned} f_{\boldsymbol{Y}}(\boldsymbol{y}) &= \frac{1}{\Gamma(1/\nu)\sqrt{2\pi}} \left(\frac{\sqrt{(\boldsymbol{y}-\boldsymbol{\delta})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}-\boldsymbol{\delta})}}{c_2} \right)^{\frac{1/\nu - d/2}{2}} e^{(\boldsymbol{y}-\boldsymbol{\delta})\Sigma^{-1}\boldsymbol{\mu}} \\ &\qquad K_{1/\nu-d/2}\left(c_2 \sqrt{(\boldsymbol{y}-\boldsymbol{\delta})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}-\boldsymbol{\delta})} \right), \end{aligned}$$

where d is the dimension of $\boldsymbol{Y}$, $c_2 = \sqrt{2 + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}$ and $K_{1/\nu-d/2}(\cdot)$ is the modified Bessel function of the second kind. The normal mean-variance mixture representation can give an intuitive feel of the distribution. This is a random variable $\boldsymbol{Y}$ having GAL distribution, and the following equality holds:

$$\boldsymbol{Y} \stackrel{d}{=} \boldsymbol{\delta} + \Gamma\boldsymbol{\mu} + \sqrt{\Gamma}\boldsymbol{\Sigma}^{1/2}\boldsymbol{Z},$$

where Γ is a Gamma distributed random variable with shape $1/\nu$ and scale one, and $\boldsymbol{Z}$ is a vector of d independent standard normal random variable. For more details see Barndorff-Nielsen et al. (1982).

2.2.2 Discrete distribution

Assume that Y_t is a univariate discrete random variable and that it has the possible values $\{1, 2, ..., M\}$. The probability distribution of observation symbols in each state is given by the emission matrix, $\boldsymbol{B} = \{b_{i,j}\}$, where

$$b_{i,j} = P(Y_t = j|Z_t = i), \ i = 1, 2, ..., N, \ j = 1, 2, ..., M$$

where $\sum_{j=1}^{M} b_{i,j} = 1$.

2.3 Estimation of HMM parameters

In our study, we mainly focus on estimating the transition matrix $\boldsymbol{Q}$ of HMM based on the observation sequences. The reason for this is that the conditional distribution of observations given a certain hidden state, describes the property of driving events which is constant over time and can be assumed to be known. However, the frequency and duration of events, which is described by the transition probabilities, are likely to depend on the type of road. Therefore, it is relevant to update the transition matrix for a new signal to find the hidden states.

We use the EM algorithm for parameter estimation. The EM algorithm is a common method for estimating the parameters in HMMs. It is an optimization algorithm to find the parameters that maximize the likelihood. The algorithm is both robust and is often easy to implement.

2.3.1 EM algorithm

The expectation-maximization (EM) algorithm, introduced by Dempster et al. (1977), is an iterative algorithm for finding the parameters that maximize the likelihood function. The EM algorithm can be used to estimate the HMM parameters. Let $\Theta = (\boldsymbol{Q}, \boldsymbol{\theta})$ denote the set of parameters in an HMM, where $\boldsymbol{Q} = (q(i,j))$ is the transition matrix of Markov chain Z_t and $\boldsymbol{\theta}$ denotes the parameters of the conditional distribution of Y_t given Z_t. Here, the sequence of hidden states Z_t takes values on a discrete space $\{1, 2, \ldots, N\}$. We assume that $y_0, ..., y_T$ are observed, and $z_0, ..., z_T$ are referred as missing data. The complete data likelihood is given by:

$$p(y_0, ..., y_T, z_0, ..., z_T; \Theta) = \pi_{z_0} g_{\boldsymbol{\theta}}(z_0, y_0) q(z_0, z_1) g_{\boldsymbol{\theta}}(z_1, y_1) ... q(z_{T-1}, z_T) g_{\boldsymbol{\theta}}(z_T, y_T). \tag{5}$$

The likelihood of interest is the joint probability density function of $Y_0, ..., Y_T$, given by:

$$L(\Theta) = p(y_0, ..., y_T; \Theta) = \sum_{z_0=1}^{N} ... \sum_{z_T=1}^{N} p(y_0, ..., y_T, z_0, ..., z_T; \Theta). \tag{6}$$

Computing the sums in the likelihood function $L(\Theta)$ is not numerically feasible. Thus, direct maximization of the likelihood is not computationally tractable. The EM algorithm provides a method for estimating the HMM parameters by using the expected value of the complete data log-likelihood given known observations and current parameters. The algorithm starts with an initial guess of the parameters $\Theta^{(0)}$, and then iteratively updates the current parameters by maximizing:

$$
\begin{aligned}
\tau(\Theta, \Theta^{(n)}) &= E\left[\log p(Y_0, ..., Y_T, Z_0, ..., Z_T; \Theta)\big|y_0, ..., y_T; \Theta^{(n)}\right] \\
&= E\left[\log \pi_{z_0}\big|y_0, ..., y_T; \Theta^{(n)}\right] \\
&+ \sum_{l=0}^{T-1} E\left[\log q(z_l, z_{l+1})\big|y_0, ..., y_T; \Theta^{(n)}\right] \\
&+ \sum_{l=0}^{T} E\left[\log g_{\boldsymbol{\theta}}(z_l, y_l)\big|y_0, ..., y_T; \Theta^{(n)}\right],
\end{aligned}
$$

for $n = 0, 1, 2, ...$ until convergence. Thus, the n^{th} iteration of the EM algorithm consists of the following two steps:

- The E-step, where the expected complete log-likelihood $\tau\left(\Theta, \Theta^{(n)}\right)$ is computed,
- The M-step, where the maximum likelihood estimate of the parameter $\Theta^{(n+1)} = \underset{\Theta}{\operatorname{argmax}}\, \tau\left(\Theta, \Theta^{(n)}\right)$ is computed.

For our specific model, the parameter of interest is $\boldsymbol{Q} = (q(i,j))$. In this case, the E-step consists of computing $P(Z_{t-1} = i, Z_t = j|y_0, ..., y_T; \boldsymbol{Q}^{(n)})$, which is the conditional probability of being at state i at time $t-1$ and state j at time t when the observation sequence and the parameters are given, and the M-step uses the conditional probability to compute the transition probabilities as follows:

$$
\begin{aligned}
q^{(n+1)}(i,j) &= \frac{\text{"}Expected\ number\ of\ transitions\ from\ state\ i\ to\ j\text{"}}{\text{"}Expected\ number\ of\ visits\ to\ state\ i\text{"}} \\
&= \frac{\sum_{l=0}^{T-1} P(Z_l = i, Z_{l+1} = j|y_0, ..., y_t; \boldsymbol{Q}^{(n)})}{\sum_{l=0}^{T} P(Z_l = i|y_0, ..., y_t; \boldsymbol{Q}^{(n)})}.
\end{aligned}
$$

2.4 Estimating the number of events

Recall that the hidden Markov chain Z_t represents the steering states. Let $\hat{\boldsymbol{Q}}$ denote the estimation of transition matrix $\boldsymbol{Q}$ computed by EM algorithm. We consider two possible ways to estimate the number of steering events:

1. The sequence of hidden states is reconstructed using $\hat{\boldsymbol{Q}}$ and the Viterbi algorithm, Viterbi (1967), and the steering events are counted.
2. The expected number of steering events are computed based on the estimated transition matrix $\hat{\boldsymbol{Q}}$.

2.4.1 Detecting the driving events

The Viterbi algorithm, see Viterbi (1967), is an important algorithm in HMMs and is used to find the most probable sequence of hidden states given the observation sequences. Suppose that we have an observation sequence $y_0, y_1, ..., y_T$ and would like to find driving events for this observation. It means that we should find a sequence of hidden states which maximizes the probability of observing this specified observation. The Viterbi algorithm determines the most likely sequence $\hat{z}_t$ of hidden states which maximizes the conditional probability of the observation sequence for given parameters Θ:

$$(\hat{z}_0, ..., \hat{z}_T) = \arg \max_{z_0,...,z_T} p(y_0, ..., y_T | z_0, ..., z_T; \Theta).$$

By using the Viterbi algorithm, we reconstruct the sequence of hidden states using the estimated parameters $\Theta = (\hat{\boldsymbol{Q}}, \boldsymbol{\theta})$ where $\hat{\boldsymbol{Q}}$ is the estimated transition matrix. In order to count the number of events, we count the number of times that $\hat{z}_t$ switches its values.

2.4.2 Expected number of events

Instead of reconstructing the sequence of driving events, it is possible to compute the expected number of events using the transition matrix. Suppose that the Markov chain $\{Z_t\}_{t=0}^{\infty}$ has transition matrix $\boldsymbol{Q}$. The expected number of i^{th} event for $\{Z_t\}_{t=0}^{T}$ is equivalent to the number of times that transitions $j \to i$ for all $j \neq i$ occur. The intensity of visiting state i is $\xi_i(t) = \sum_{j \neq i} I(Z_{t-1} = j, Z_t = i)$, for $t = 1, 2, ..., T$. In addition, one should consider the state at time zero, Z_0, which can also be i. Therefore, the expected number of i^{th} event is:

$$\eta_i = E[I(Z_0 = i)] + E[\sum_{t=1}^{T} \xi_i(t)] = \pi_i + T \sum_{j \neq i} \pi_j q(j, i), \; i, j = 1, 2, ..., N \tag{7}$$

where $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_N)$ is the stationary distribution of $\boldsymbol{Q}$ computed by solving equation $(\boldsymbol{Q} - I)\boldsymbol{\pi} = \boldsymbol{0}$.
The expected number of events is estimated by replacing $\boldsymbol{Q}$ by $\hat{\boldsymbol{Q}}$.

3 HMM for steering events

The steering events are important since they generate high forces in the steering components. We divide the steering events into curves and maneuvers. The curve events give rise to lateral forces through lateral acceleration and occur when driving at speeds higher than about 10 km/h, while the maneuvering events generate high forces due to steering at low speeds, typically lower than 10 km/h, e.g. driving in or out of a parking lot, standing still but turning steering wheel.

We use a single HMM to detect both the curves and maneuvers. In order to identify the events, we consider six states for the hidden Z-process as follows:

right turn (1 = "RT"), straight forward for turns (2 = "SFT") and left turn (3 = "LT"), steering right (4 = "SR"), straight forward for maneuvers (5 = "SFM") and steering left (6 = "SL"). Three signals, lateral acceleration ($Y_{t,1}$), steering angle speed ($Y_{t,2}$) and vehicle speed ($Y_{t,3}$) are used as observation in the HMM. We assume that $Y_{t,1}$ and $Y_{t,2}$ are continuous observations and we discretized $Y_{t,3}$ into the three levels:

- $1 = \{0 \text{ km/h} \leq \text{"}speed\text{"} < 1 \text{ km/h}\}$,
- $2 = \{1 \text{ km/h} \leq \text{"}speed\text{"} < 10 \text{ km/h}\}$,
- $3 = \{\text{"}speed\text{"} \geq 10 \text{ km/h}\}$.

We use a combination of Laplace and discrete distributions for observations and construct an HMM to identify the steering events.

4 Vehicle logging data

In this section we illustrate the on-board logging signals available on trucks. This information is available for all vehicles and can be obtained from CAN (Controller Area Network) bus data. There are more than 80 signals from CAN bus data that can be used to identify the driving events. The signals we have used from CAN-data are named:

- Steering wheel angles,
- Vehicle speed,
- Yaw rate.

The steering wheel is the wheel the driver holds in his or her hand while driving and the angle is defined as the angle deviation from driving straight ahead.

The yaw rate contains information about steering events only for a non-zero speed. By steering, the entire vehicle will shift direction and this will happen at a certain angular velocity, which is the yaw rate.

We have used the yaw rate and speed to get an accurate lateral acceleration signal, which is computed by the following formula:

$$\text{"}lateral\ acceleration\text{"} = \text{"}speed\text{"} \cdot \text{"}yaw\ rate\text{"}.$$

The lateral force is proportional to the lateral acceleration in turning events. For most components these loads are not as damaging as the vertical loads, but they have a large impact on steering components, Karlsson (2006).

5 Example

Recall that the aim of this study is to estimate the number of events in measurements. We use a single HMM to identify both the curves and the maneuvers. The number of curves and maneuvers are estimated by two different methods. In the first approach, the estimated transition matrix and the Viterbi algorithm are used to reconstruct the most likely sequence $\hat{z}_t$ of steering states and then the number of events is counted. In the second approach, the expected number of events are estimated using the estimated transition matrix and Eq. (7).

A measured data set is used to demonstrate the proposed algorithm. In a second study, a number of simulated data sets are used to check how well the estimation algorithm works. We especially assess the statistical properties of the estimators, such as bias and variance.

5.1 Measured data

As mentioned above, a real data set is used to demonstrate the algorithm. We use the dedicated field measurements from a Volvo Truck as our data set. We divide the data into two equal portions as our training and test sets. We use the training set to estimate the parameters of the model and the test set to validate the model.

In our case study, the training set contains all history about the steering events such as the start and stop points. By using this information of the training set, the parameter $\boldsymbol{\theta}$ of the conditional distribution of observation given the hidden state is found through maximum likelihood estimation.

We use an HMM to re-estimate the transition matrix $\boldsymbol{Q}$ and to detect the steering events from the test set. The estimated parameters from the training set are used for the conditional distribution of observations. This is because the conditional distribution of Y_t given a certain hidden state Z_t describes the property of driving events which is assumed to be vehicle specific data that can be estimated under well-defined conditions on the proving ground. However, the differences between types of roads can affect the transition probabilities and the duration of the events. Therefore, it could be relevant to update the transition matrix $\boldsymbol{Q}$ of HMM based on a new observation sequence.

5.1.1 Training set

To create training data, the steering events are detected manually by looking at video recordings from the truck cabin. Then, the start and stop points of each event are extracted to create the hidden Z-process. As mentioned above, three signals: lateral acceleration, steering angle speed and the discretized speed are the observations in our HMM. Figure 1 shows the training data which contain the measured lateral acceleration, steering wheel angles, vehicle speed and discretized vehicle seed with the corresponding manually detected steering states. This training set has been used to estimate all parameters of the model.

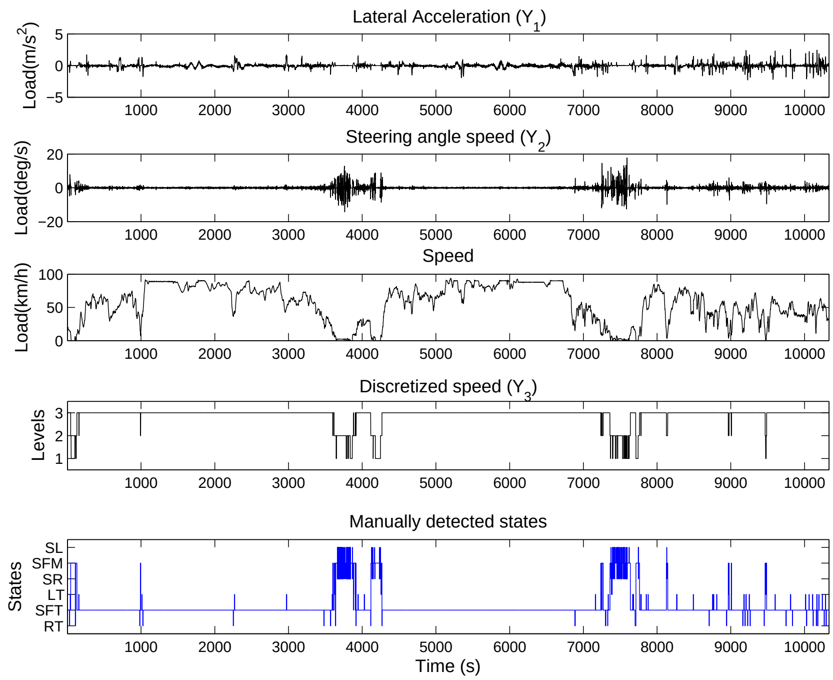


Figure 1: The training data contain the measured lateral acceleration (Y_1), steering angle speed (Y_2) and vehicle speed (Y_3) with the corresponding manually detected steering states.

We have two choices for modeling observations Y_1 and Y_2 with the Laplace distribution. We can either consider that Y_1 and Y_2 are dependent, which means that they have the same shape parameter ν, or we can assume that they are independent but with different ν. Both cases have been tested. It was found that having separate ν for Y_1 and Y_2 gives better fit to data and also better estimation of the transition matrix. Thus, we assume that Y_1 and Y_2 are independent variables, each having a Laplace distribution. The histograms and the fitted distributions for lateral acceleration and steering angle speed for each state are shown in Figure 2.

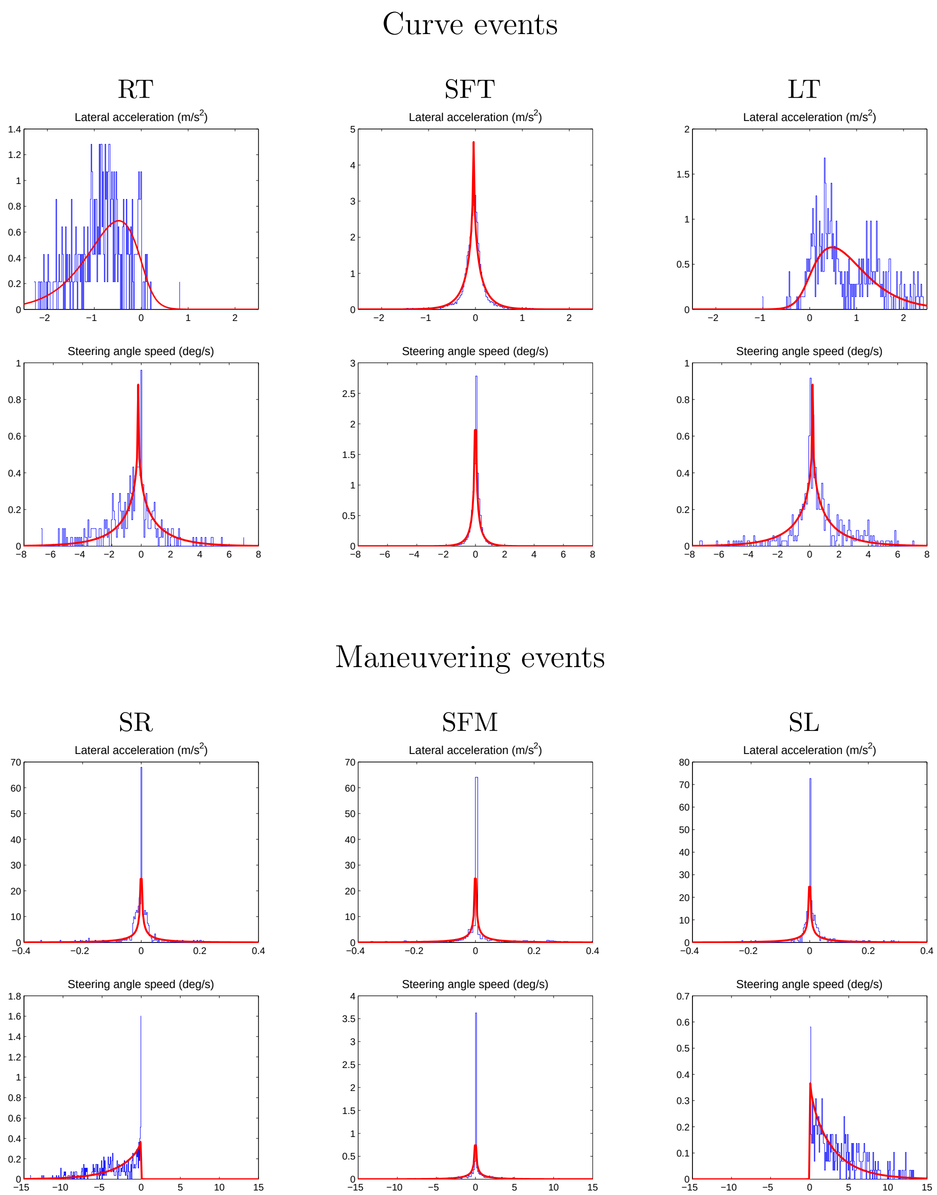


Figure 2: The histogram of observation and the fitted Laplace distribution for curves and maneuvers.

The parameter of the Laplace distributions given the hidden state, is found through maximum likelihood estimation. The estimated parameters for steering states are given in Table 1.

Table 1: The estimated parameters of Laplace distributions fitted for lateral acceleration (Y_1) and steering angle speed (Y_2).

	State(i)	1="RT"	2="SFT"	3="LT"	4="SR"	5="SFM"	6="SL"
Y_1	δ_i	0.04	-0.04	-0.04	3.02×10^{-6}	-5.59×10^{-6}	-3.02×10^{-6}
	μ_i	-0.26	-8.21×10^{-4}	0.26	-0.003	0.003	0.003
	ν_i	0.31	1.73	0.31	6.17	7.56	6.17
	σ_i^2	0.07	0.13	0.07	0.04	0.04	0.04
Y_2	δ_i	-0.20	1.36×10^{-5}	0.20	7.63×10^{-4}	-3.81×10^{-5}	-7.63×10^{-4}
	μ_i	-0.01	0.01	0.01	-3.27	0.02	3.27
	ν_i	1.78	3.33	1.78	1.07	5.75	1.07
	σ_i^2	7.54	0.71	7.54	0.08	23.73	0.08

The estimated probabilities of discretized speed (Y_3) in each state is given by the emission matrix $\boldsymbol{B}$:

$$\boldsymbol{B} = \begin{array}{c} \\ \text{RT} \\ \text{SFT} \\ \text{LT} \\ \text{SR} \\ \text{SFM} \\ \text{SL} \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0.13 & 0.87 \\ 0 & 0 & 1 \\ 0 & 0.21 & 0.79 \\ 0.17 & 0.83 & 0 \\ 0.44 & 0.56 & 0 \\ 0.22 & 0.78 & 0 \end{pmatrix} \end{array}.$$

5.1.2 Test set

Here, we evaluate the proposed method using the test set. We use the estimated parameters for conditional distribution of observations from the training set, and estimate the transition matrix $\boldsymbol{Q}$ based on the test set for detecting the events in the test set. The EM algorithm is used to estimate the transition matrix $\boldsymbol{Q}$, and the estimated matrix is:

$$\hat{\boldsymbol{Q}} = \begin{array}{c} \\ \text{RT} \\ \text{SFT} \\ \text{LT} \\ \text{SR} \\ \text{SFM} \\ \text{SL} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{RT} & \text{SFT} & \text{LT} & \text{SR} & \text{SFM} & \text{SL} \end{array} \\ \begin{pmatrix} 0.9274 & 0.0574 & 0.0108 & 0.0005 & 0.0006 & 0.0032 \\ 0.0082 & 0.9834 & 0.0075 & 0.0001 & 0.0003 & 0.0004 \\ 0.0117 & 0.0661 & 0.9190 & 0.0032 & 0.0000 & 0.0000 \\ 0.0131 & 0.0100 & 0.0000 & 0.8369 & 0.0198 & 0.1202 \\ 0.0000 & 0.0037 & 0.0002 & 0.0231 & 0.9501 & 0.0228 \\ 0.0008 & 0.0000 & 0.0147 & 0.1023 & 0.0441 & 0.8380 \end{pmatrix} \end{array}.$$

By using the estimated transition matrix and Viterbi algorithm, we reconstruct the most likely sequence $\hat{z}_t$ of steering states to find the frequency of events. The estimated number of events from the Viterbi algorithm are given by $\hat{\kappa}_i$. Further, we estimate the expected number of events, $\hat{\eta}_i$, calculated from the estimated transition matrix using Eq. (7). The results are presented in Table 2.

Table 2: The estimated number of events $\hat{\kappa}_i$ from the Viterbi algorithm and the expected number of events $\hat{\eta}_i$ using estimated transition matrix $\hat{Q}$.

Event(i)	1="RT"	2="SFT"	3="LT"	4="SR"	5="SFM"	6="SL"
$\hat{\kappa}_i$	137	220	137	96	37	101
$\hat{\eta}_i$	150	245	143	95	46	104

One can conclude that the expected number of right and left turns estimated from the transition matrix $\hat{Q}$ is larger than the number of turns found by the Viterbi algorithm. However there is no significant difference between the number of maneuvers estimated from both methods. Figure 3 shows the testing data and the corresponding detected states based on HMM and Viterbi algorithm.

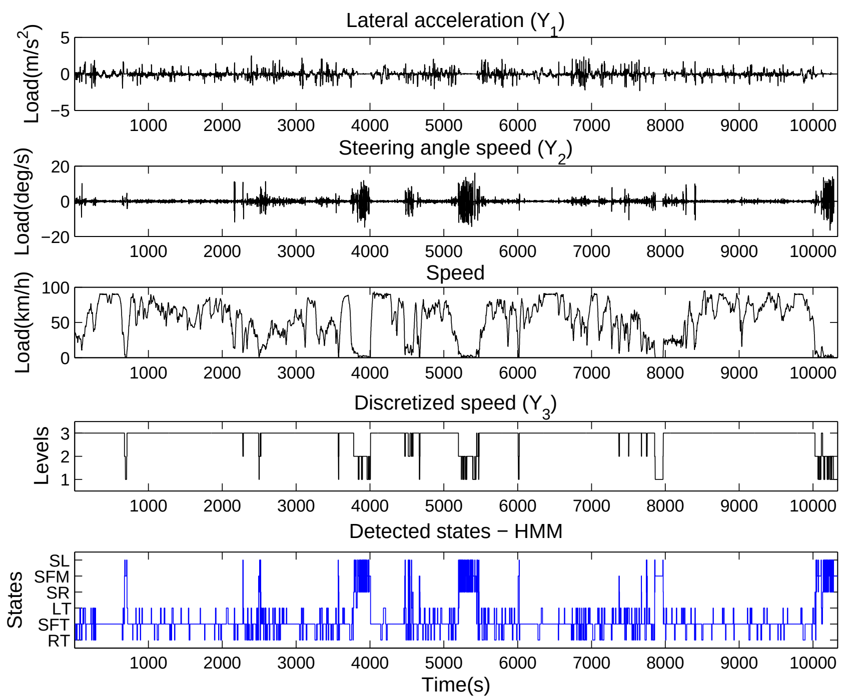


Figure 3: The testing data contain the measured lateral acceleration (Y_1), steering angle speed (Y_2) and vehicle speed (Y_3) with the corresponding steering states detected by HMM.

5.2 Simulated data

The goal of the simulation study is to investigate the properties of the estimates of the number of events. We use two approaches to estimate the number of steering events. In the first approach, we use the Viterbi algorithm to count the number of events. In the second approach, we estimate the expected number of events using the estimated transition matrix $\hat{Q}$. We investigate the properties of these two different estimates through a simulation study. In order to assess the performance of the algorithm, we estimate four different errors, which are defined in the following table.

κ_i	Observed number of i^{th} event for $i = 1, 2, ..., 6$
$\hat{\kappa}_i$	Estimated number of i^{th} event from the Viterbi algorithm
η_i	Expected number of i^{th} event using true transition matrix $\boldsymbol{Q}$
$\hat{\eta}_i$	Estimate of η_i using estimated transition matrix $\hat{\boldsymbol{Q}}$
$e_{1,i}$	Error1 $= \hat{\kappa}_i - \kappa_i$
$\epsilon_{1,i}$	Error2 $= \hat{\kappa}_i - \eta_i$
$e_{2,i}$	Error3 $= \hat{\eta}_i - \kappa_i$
$\epsilon_{2,i}$	Error4 $= \hat{\eta}_i - \eta_i$

We first generate the sequence of steering states by using a Markov chain. We construct an HMM based on six steering states with the transition matrix:

$$\boldsymbol{Q} = \begin{array}{c} \\ \text{RT} \\ \text{SFT} \\ \text{LT} \\ \text{SR} \\ \text{SFM} \\ \text{SL} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{RT} & \text{SFT} & \text{LT} & \text{SR} & \text{SFM} & \text{SL} \end{array} \\ \begin{pmatrix} 0.90 & 0.09 & 0.005 & 0 & 0.005 & 0 \\ 0.0225 & 0.95 & 0.0225 & 0 & 0.005 & 0 \\ 0.005 & 0.09 & 0.90 & 0 & 0.005 & 0 \\ 0 & 0.005 & 0 & 0.90 & 0.09 & 0.005 \\ 0 & 0.005 & 0 & 0.0225 & 0.95 & 0.0225 \\ 0 & 0.005 & 0 & 0.005 & 0.09 & 0.90 \end{pmatrix} \end{array}.$$

Three signals: lateral acceleration ($Y_{t,1}$), steering angle speed ($Y_{t,2}$) and vehicle speed ($Y_{t,3}$) are simulated as observation sequences. We assume that $Y_{t,1}$ and $Y_{t,2}$ are independent random variables with a Laplace distribution. We simulate the lateral acceleration and steering angle speed where the parameters are given in Table 3.

Table 3: The parameters of Laplace distributions for lateral acceleration (Y_1) and steering angle speed (Y_2).

	State(i)	1="RT"	2="SFT"	3="LT"	4="SR"	5="SFM"	6="SL"
Y_1	δ_i	-1	0	1	0	0	0
	μ_i	-0.5	0	0.5	-0.003	0.003	0.003
	ν_i	0.1	2	0.1	5	8	5
	σ_i^2	0.2	1	0.2	0.04	0.04	0.04
Y_2	δ_i	-0.4	0	0.4	-0.003	0	0.003
	μ_i	-0.01	0.01	0.01	-5	0.004	5
	ν_i	2	3	2	1	5	1
	σ_i^2	7	0.6	7	0.2	10	0.2

The vehicle speed is considered a discrete observation with three levels $\{1, 2, 3\}$, as described on page 8. We simulate a sequence of discretized speed ($Y_{t,3}$) by using the following emission matrix:

$$\boldsymbol{B} = \begin{array}{l} \\ \text{RT} \\ \text{SFT} \\ \text{LT} \\ \text{SR} \\ \text{SFM} \\ \text{SL} \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0.2 & 0.8 \\ 0 & 0.2 & 0.8 \\ 0 & 0.2 & 0.8 \\ 0.2 & 0.8 & 0 \\ 0.2 & 0.8 & 0 \\ 0.2 & 0.8 & 0 \end{pmatrix} \end{array}.$$

The simulated signals represent a journey on a city road over 10^4 seconds (≈ 3 hours), where the sampling period is 1/2 seconds. We perform 1000 simulations of city road to estimate the number of steering events. Since we consider the parameters corresponding to the conditional distribution of observations given a hidden state as fixed, we use the EM algorithm to estimate the transition matrix given three simulated signals as observation process $\boldsymbol{Y}_t = (Y_{t,1}, Y_{t,2}, Y_{t,3})$.

For each simulation, the number of events has been estimated based on two approaches. In order to investigate the properties of the estimators, we compute four different errors. The first two errors $e_{1,i}$ and $\epsilon_{1,i}$ correspond to the first approach, while $e_{2,i}$ and $\epsilon_{2,i}$ are computed based on the second approach. Errors $e_{1,i}$ and $\epsilon_{1,i}$ represent how much the estimated number of events from the Viterbi algorithm ($\hat{\kappa}_i$) differs from the observed number of events (κ_i) and the expected number of events (η_i), respectively. Error $e_{2,i}$ is the difference between the estimated expected number of events ($\hat{\eta}_i$) and the observed number of events, while $\epsilon_{2,i}$ indicates the difference between the estimated and true values of the expected number of events. We compute the mean and the standard deviation of errors over 1000 simulations. The 95% confidence intervals are indicated in the parenthesis. The results are presented in Table 4.

Table 4: The estimated number of steering events. The mean and the standard deviation of errors with the 95% confidence intervals.

Event(i)	1="RT"	2="SFT"	3="LT"	4="SR"	5="SFM"	6="SL"
mean(κ_i)	81	172	82	80	168	79
std(κ_i)	13.2	22.9	13.4	13.6	23.1	14
η_i	80	170	80	80	170	80
mean($\hat{\kappa}_i$)	84	178	83	76	156	76
std($\hat{\kappa}_i$)	13.6	23.2	13.5	13.5	22.2	13.8
mean($\hat{\eta}_i$)	81	172	82	83	174	82
std($\hat{\eta}_i$)	13.4	23.1	13.5	15.1	25.2	15.5
mean($e_{1,i}$)	2.8 ($\pm$0.1)	5.8 ($\pm$0.2)	0.9 ($\pm$0.1)	-3.2 ($\pm$0.3)	-11.8 ($\pm$0.5)	-3.2 ($\pm$0.4)
std($e_{1,i}$)	2.4	3.3	1.8	5.5	8.1	5.6
mean($\epsilon_{1,i}$)	3.7 ($\pm$0.9)	7.9 ($\pm$1.5)	2.3 ($\pm$0.9)	-4 ($\pm$0.9)	-14 ($\pm$1.4)	-4.3 ($\pm$0.9)
std($\epsilon_{1,i}$)	13.6	23.2	13.5	13.6	22.2	13.8
mean($e_{2,i}$)	0.01 ($\pm$0.1)	0.01 ($\pm$0.2)	0.02 ($\pm$0.1)	2.9 ($\pm$0.3)	6.3 ($\pm$0.5)	3 ($\pm$0.3)
std($e_{2,i}$)	1.8	3	1.8	5.3	7.9	5.4
mean($\epsilon_{2,i}$)	0.9 ($\pm$0.8)	2.1 ($\pm$1.5)	1.3 ($\pm$0.8)	2.1 ($\pm$0.9)	4.1 ($\pm$1.6)	1.8 ($\pm$1.0)
std($\epsilon_{2,i}$)	13.4	23.1	13.5	15.1	25.2	15.5
corr($\hat{\kappa}_i, \kappa_i$)	0.98	0.99	0.99	0.92	0.94	0.92
corr($\hat{\eta}_i, \kappa_i$)	0.99	0.99	0.99	0.94	0.95	0.94

Approach one with errors $e_{1,i}$ and $\epsilon_{1,i}$ gives bias for estimation of κ_i and η_i, respectively. As can be seen from the 95% confidence intervals, we have positive bias for turns and negative bias for maneuvers based on the first approach. According to the second approach and error $e_{2,i}$, there is only bias for maneuvers, while error $\epsilon_{2,i}$ indicates a small bias.

The standard deviations of the errors calculated from both approaches are almost the same, while the bias of the second approach is smaller than the first one. Therefore, the second method should be preferred to estimate the number of events.

It can also be seen that κ_i is highly correlated with $\hat{\kappa}_i$ and $\hat{\eta}_i$, which is expected since the standard deviations of $e_{1,i}$ and $e_{2,i}$ are small.

6 Conclusion

A method is proposed to identify steering events (curves and maneuvers), using an HMM with multivariate observation sequences. We considered six driving states: right turn (RT), straight forward for turns (SFT), left turn (LT), steering right (SR), straight forward for maneuvers (SFM) and steering left (SL), to construct the HMM. Three signals: lateral acceleration, steering angle speed and speed are used as observations. It is shown that hidden Markov models with a combination of continuous and discrete distributions for observations can be used to find the steering events. Both simulated and measured signals are used to exemplify the identification of steering events. Two different approaches have been used to estimate the number of events. In the first approach, the number of events have been counted based on the Viterbi algorithm. In the second approach, we estimate the expected number of events from the transition matrix estimated by the EM algorithm. The bias and variance of estimations are investigated. The standard deviations of the errors computed from both approaches are almost the same, while the bias of the second approach is smaller than the first one. Therefore, one can conclude that the second approach, using the expected number of events, can accurately estimate the number of steering events and outperform the Viterbi algorithm.

The proposed model can be extended for detecting more steering events such as static steering or slow reverse steering. This can be done by increasing the number of driving states and also the number of observation sequences in the HMM. For instance, to distinguish between forward and reverse maneuvers, we need to use a current gear signal as an extra observation.

Acknowledgment

We would like to thank Volvo Trucks for supplying the data in this study and the members in our research group at Volvo for their valuable advice. We gratefully acknowledge the financial support from VINNOVA.

References

Barndorff-Nielsen, O., Kent, J., and Sorensen, M. (1982). Normal variance-mean mixtures and z distributions. *International Statistical Review*, 50:145–159.

Berndt, H. and Dietmayer, K. (2009). Driver intention inference with vehicle onboard sensors. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 102–107, Pune.

Cappé, O., Moulines, E., and Rydén, T., editors (2005). *Inference in Hidden Markov Models*. Springer.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Johannesson, P. and Speckert, M., editors (2013). *Guide to Load Analysis for Durability in Vehicle Engineering*. Wiley:Chichester.

Karlsson, M. (2006). Statistical parameterization of lateral vehicle loads. *Vehicle System Dynamics*, 44:770–779.

Karlsson, M. (2007). *Load Modelling for Fatigue Assessment of Vehicles – a Statistical Approach*. PhD thesis, Chalmers University of Technology, Sweden.

Maghsood, R. and Johannesson, P. (2013). Detection of the curves based on lateral acceleration using hidden Markov models. *Procedia Engineering*, 66:425–434.

Maghsood, R. and Johannesson, P. (2016). Detection of steering events based on vehicle logging data using hidden Markov models. *International Journal of Vehicle Design*.

Maghsood, R., Rychlik, I., and Wallin, J. (2015). Modeling extreme loads acting on steering components using driving events. *Probabilistic Engineering Mechanics*, 41:1320.

Mitrović, D. (2004). *Learning Driving Patterns to Support Navigation*. PhD thesis, University of Canterbury, New Zealand.

Mitrović, D. (2005). Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):198–205.

Nilsson, T., Nyberg, P., Sundström, C., Frisk, E., and Krysander, M. (2014). Robust driving pattern detection and identification with a wheel loader application. *International Journal of Vehicle Systems Modelling and Testing*, 9(1):56–76.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13(2):260–269.

Paper V

On-line Estimation of Driving Events and Fatigue Damage on Vehicles

R. Maghsood and J. Wallin

On-line estimation of driving events and fatigue damage on vehicles

R. Maghsood* and J. Wallin
Chalmers University of Technology
*Corresponding author

Abstract

Driving events, such as maneuvers at slow speed and turns, are important for durability assessments of vehicle components. By counting the number of driving events, one can estimate the fatigue damage caused by the same kind of events. Through knowledge of the distribution of driving events for a group of customers, the vehicles producers can tailor the design, of vehicles, for the group. In this article, we propose an algorithm that can be applied on-board a vehicle to on-line estimate the expected number of driving events occurring, and thus be used to estimate the distribution of driving events for a certain group of customers. Since the driving events are not observed directly, the algorithm uses a hidden Markov model to extract the events. The parameters of the HMM are estimated using an on-line EM algorithm, with a fixed forgetting factor to address chaining driving conditions.

Keywords: Hidden Markov models; EM algorithm; on-line EM algorithm; driving events; expected damage; fatigue damage.

1 Introduction

When designing vehicles components it is important to know the distributions of loads expected to act on them. The life time of a component in a vehicle– such as control arms, ball joints, etc.– is determined by its strength and the loads acting on it. Where the effect of a given force acting on a component is well known, the distributions of loads, and hence forces, are more random. This is because the distribution of the loads depends on the driving environment, driver's behavior, usage of the vehicle, and other things. For a more detailed description of loads acting on vehicles, see Johannesson and Speckert (2013).

Although it is not financially possible to design a vehicle for specific customer, it is important to tailor the design for groups of customers, depending on, for instance, geographical regions and usage. Obviously, components weakly designed for the specific environments leads to increased costs due to call-backs and badwill for the company, while too heavily designed components give increased material cost and unnecessarily heavy vehicles.

Traditionally, one has used a specially equipped test vehicle to study the distributions of customer loads. This gives very precise measurements, but with disadvantage of a statistically small sample size for the studied group. In addition, it is a very expensive way of acquiring data. However, all modern vehicles are equipped with computers measuring many signals, known as Controller Area Network (CAN) bus data, where the signal is for instance speed and lateral acceleration. The goal of this article is to develop a statistical algorithm that uses these signals, to extract information about the driving events for the specific vehicle. This data can then be collected from several vehicles to generate a load distribution for groups of customers.

The desired algorithm needs several key properties to be practically useful: First, it obviously needs to be able to extract the driving events from the CAN data. Second, since the data will be extracted over long periods of time the computational cost of estimation of the driving events needs to be low. It is also desirable that the method does not require the storage of all the data. Finally, the algorithm should allow for changing frequency of driving events over time, since the frequency of driving events changes depending on the driving environment such as highway driving or city driving.

To address the first property, our algorithm uses a hidden Markov Model (HMM) to extract the driving events from the CAN data. More specifically each state in the HMM represents a driving state where we define a driving event as a sequence of consecutive driving states. The CAN data for a given driving state is assumed to follow a generalized Laplace (GAL) distribution. Laplace distributions are well known methods to describe responses measured on driving vehicles, see Bogsjö et al. (2012); Kvanström et al. (2013). The idea of using HMMs to identify driving events has previously been used in for example Maghsood and Johannesson (2013, 2016), Mitrović (2004, 2005) and Berndt and Dietmayer (2009).

For the HMM we divide the parameter into two sets: the transition matrix, which is vehicle type independent, depending rather on the driving environment, the driver's behavior etc. The parameter of the GAL distribution is vehicle type specific, and can thus be found in laboratory tests or in proving grounds. Thus the second property, in the case of an HMM, is equivalent to efficiently estimating the transition matrix of driving states. In previous articles, the EM algorithm has been used successfully to estimate the transition matrix, Maghsood et al. (2015); however an iteration of the algorithm has computational complexity $\mathcal{O}(n)$ (where n is the number of observation) and is thus not practically feasible. Here, we instead propose using the on-line EM algorithm from Cappé (2011) to estimate the matrix. This gives the desired computational efficiency, since one iteration of the algorithm has a computational cost of $\mathcal{O}(1)$.

The final property is addressed by using a fixed forgetting factor in the on-line EM algorithm. Cappé (2011) proposes a diminishing forgetting factor to ensure that the EM algorithm converges to a stationary point. However, this is not the goal here and we do not want the algorithm to converge to a stationary point but rather be an adaptive algorithm. The usage of forgetting factors is a well-studied area in automatic control, time series analysis and vehicle engineering, see Arvastson et al. (2000); Lennart and Söderström (1983); Vahidi et al. (2005).

Further, the algorithm also calculates on-line the expected damage for a given component. This could be useful for the specific vehicle, on which the algorithm is applied, by using the expected damage to tailor service times to specific vehicle and components.

The paper is organized as follows: In the second section, the HMM and the proposed on-line algorithm are presented. In the third section, the method for estimating the fatigue damage is proposed. In the forth section, the algorithm is applied to simulated data to verify its performance, and it is also evaluated on real data, CAN data from a Volvo truck. The final section contains the conclusions of the paper.

2 Hidden Markov models

Hidden Markov models are statistical models often used in signal processing, such as speech recognition and modeling the financial time series, see for instance Cappé et al. (2005) and Frühwirth-Schnatter (2006). An HMM is a bivariate Markov process $\{Z_t, Y_t\}_{t=0}^{\infty}$ where the underlying process Z_t is an unobservable Markov chain and is observed only through the Y_t. The observation sequence Y_t given Z_t is a sequence of independent random variables and the conditional distribution of Y_t depends only on Z_t.

In this article, all HMMs are such that Z_t takes values on a discrete space $\{1, 2, \dots, m\}$, and the HMM is determined by two sets of parameters. The first set is the transition probabilities of Markov chain Z_t:

$$q(i,j) = P(Z_{t+1} = j | Z_t = i), \; i,j = 1,2,...,m. \tag{1}$$

The second set is the parameter vector, $\boldsymbol{\theta}$, of the conditional distribution of Y_t given Z_t:

$$g_{\boldsymbol{\theta}}(i, y_t) = f_{Y_t}(y_t | Z_t = i; \boldsymbol{\theta}), \; i = 1,2,...,m, \; y_t \in \mathbb{R}. \tag{2}$$

Here, we denote the set of parameters by $\Theta = (\boldsymbol{Q}, \boldsymbol{\theta})$ where $\boldsymbol{Q} = (q(i,j))$ for $i,j = 1,2,...,m$.

In an HMM, the state where the hidden process will start is modeled by the initial state probabilities $\boldsymbol{\pi} = (\pi_i)$, where π_i is denoted by:

$$\pi_i = P(Z_0 = i), \; i = 1,2,...,m$$

with $\sum_{i=1}^{m} \pi_i = 1$.

2.1 Parameter estimation

For the parameter estimation in this article we use the EM (expectation maximization) algorithm, which is described below. The principle aim is to estimate the transition matrix $\boldsymbol{Q}$ based on an observation sequence. For this, we use an on-line EM algorithm, derived in Cappé (2011). To introduce the algorithm we first describe the EM algorithm and then describe the modification needed for on-line usage of the algorithm.

In our study, the parameter $\boldsymbol{\theta}$ is not estimated recursively, but rather found through maximum likelihood estimation on a training set. This is because the conditional distribution of Y_t given Z_t in our case study represents the vehicle specific data which can be estimated under well-defined conditions on the proving ground.

2.2 The EM algorithm

Here, we present the EM algorithm following Cappé (2011). The EM algorithm is a common method for estimating the parameters in HMMs. It is an optimization algorithm to find the parameters that maximize the likelihood. The algorithm is both robust – it does not diverge easily– and is often easy to implementation.

The EM algorithm is an iterative procedure. If the distribution of complete-data (Z_t, Y_t) given Z_{t-1}, $p(z_t, y_t | z_{t-1})$, belongs to an exponential family, then the n^{th} iteration consists of the two following steps:

- The E-step, where the conditional expectation of the complete-data sufficient statistics, $s(Z_{t-1}, Z_t, Y_t)$, given the observation sequence $y_0, y_1, ..., y_t$ and $\Theta^{(n)}$, is computed,

$$\boldsymbol{S}_t^{(n+1)} = \frac{1}{t} E\left[\sum_{l=1}^{t} s(Z_{l-1}, Z_l, Y_l) \middle| y_0, ..., y_t; \Theta^{(n)}\right], \tag{3}$$

- The M-step, where the new parameter value $\Theta^{(n+1)}$ is calculated using $\boldsymbol{S}_t^{(n+1)}$, which can be formulated as $\Theta^{(n+1)} = f(\boldsymbol{S}_t^{(n+1)})$.

The sequence $\Theta^{(n)}$ converges to a stationary point of the likelihood function, for more details see Cappé (2011).

For our specific model, where the parameter of interest is $\boldsymbol{Q} = (q(i,j))$, the sufficient statistics will be $I(Z_{t-1} = i, Z_t = j)$ where $I(\cdot, \cdot)$ is the indicator function. Then, the E-step is:

$$S_t^{(n+1)}(i,j) = \frac{1}{t} E\left[\sum_{l=1}^{t} I(Z_{l-1} = i, Z_l = j) \middle| y_0, ..., y_t; \Theta^{(n)}\right]. \tag{4}$$

Thus $S_t(i,j)$ is the expected number of transitions from state i to state j given $y_0, ..., y_t$ and Θ. For $\boldsymbol{Q} = (q(i,j))$, the M-step is given by:

$$q^{(n+1)}(i,j) = \frac{S_t^{(n+1)}(i,j)}{\sum_{j=1}^{m} S_t^{(n+1)}(i,j)}. \tag{5}$$

2.2.1 Recursive formulation of the E-step

Zeitouni and Dembo (1988) noted that the conditional expectation of the complete-data sufficient statistics $\boldsymbol{S}_t$ can be computed recursively. To see this, define:

$$\phi_t(k) = P(Z_t = k|y_0, ..., y_t; \Theta), \tag{6}$$

$$\rho_t(i,j,k) = \frac{1}{t} E\left[\sum_{l=1}^{t} I(Z_{l-1} = i, Z_l = j)\middle| y_0, ..., y_t, Z_t = k; \Theta\right], \tag{7}$$

then $S_t(i,j)$ can be written as $S_t(i,j) = \sum_{k=1}^{m} \phi_t(k)\rho_t(i,j,k)$.

Note that $(\boldsymbol{\phi}_t)_k = \phi_t(k)$ is an N-dimensional (row) vector. For a vector $\boldsymbol{a}$, let $\boldsymbol{D}(\boldsymbol{a})$ be a diagonal matrix where $\boldsymbol{D}(\boldsymbol{a})_{kk} = a_k$. The recursive implementation of the EM algorithm, using the observation sequence $y_0, y_1, ..., y_T$, is initialized with

$$\boldsymbol{\phi}_0 = \frac{\boldsymbol{\pi D}(g_{\boldsymbol{\theta}}(k, y_0))}{(\boldsymbol{\pi D}(g_{\boldsymbol{\theta}}(k, y_0)))\mathbf{1'}}, \text{ and } \rho_0(i,j,k) = 0,$$

for all $1 \le i,j,k \le m$. Let $g_{\boldsymbol{\theta}}(y_t) = (g_{\boldsymbol{\theta}}(1, y_t), g_{\boldsymbol{\theta}}(2, y_t), ..., g_{\boldsymbol{\theta}}(m, y_t))$ and $\mathbf{1} = (1, 1, ..., 1)$. Then, for n^{th} iteration and $t \ge 1$, the components are updated as follows:

$$\boldsymbol{\phi}_{t+1} = \frac{\mathbf{1}(\boldsymbol{D}(\boldsymbol{\phi}_t)\boldsymbol{Q}^{(n)}\boldsymbol{D}(g_{\boldsymbol{\theta}}(y_t)))}{\mathbf{1}(\boldsymbol{D}(\boldsymbol{\phi}_t)\boldsymbol{Q}^{(n)}\boldsymbol{D}(g_{\boldsymbol{\theta}}(y_t)))\mathbf{1'}}, \tag{8}$$

$$\rho_{t+1}(i,j,k) = \gamma_{t+1} I(j-k) r_{t+1}(i|j) + (1-\gamma_{t+1}) \sum_{k'=1}^{m} \rho_t(i,j,k') r_{t+1}(k'|k), \tag{9}$$

where $\boldsymbol{r}_{t+1} = \boldsymbol{D}(\boldsymbol{\phi_t}./\mathbf{1}(\boldsymbol{D}(\boldsymbol{\phi}_t)\boldsymbol{Q}^{(n)}))\boldsymbol{Q}^{(n)}$ and ./ represents the element-wise division of two matrices. The forgetting factor, γ_t, equals $1/t$.

Note that in n^{th} iteration of EM algorithm, all elements in $\phi_1, \phi_2, ..., \phi_t$ and $\rho_1, \rho_2, ..., \rho_t$ depend on $\boldsymbol{Q}^{(n)}$. Thus, for updating $\boldsymbol{Q}$ in $(n+1)^{th}$ iteration, all elements of the two quantities need to be recalculated. Therefore one needs to store the entire observation vector to use the EM algorithm.

2.3 On-line estimation of HMM parameters

As we will see soon, the on-line EM algorithm remedies the issue of requiring the entire observation vector to estimate parameters. Here we use the notation $\hat{\boldsymbol{Q}}_t$ rather than $\boldsymbol{Q}^{(t)}$. This is because, as we will see, one can not compute more than one iteration at each time point t for the on-line EM.

The terms $\hat{\boldsymbol{\phi}}_0$ and $\hat{\rho}_0(i,j,k)$ are initialized the same way as in the regular EM algorithm. For $t = 0, 1, \ldots$ the components are updated as follows: (the

E-step)

$$\hat{\boldsymbol{\phi}}_{t+1} = \frac{\mathbf{1}(\boldsymbol{D}(\boldsymbol{\phi}_t)\hat{\boldsymbol{Q}}_t\boldsymbol{D}(g_{\boldsymbol{\theta}}(y_t)))}{\mathbf{1}(\boldsymbol{D}(\boldsymbol{\phi}_t)\hat{\boldsymbol{Q}}_t\boldsymbol{D}(g_{\boldsymbol{\theta}}(y_t)))\mathbf{1'}}, \tag{10}$$

$$\hat{\rho}_{t+1}(i,j,k) = \gamma_{t+1}I(j-k)\hat{r}_{t+1}(i|j) + (1-\gamma_{t+1})\sum_{k'=1}^{m}\hat{\rho}_t(i,j,k')\hat{r}_{t+1}(k'|k), \tag{11}$$

where $\hat{\boldsymbol{r}}_{t+1} = \boldsymbol{D}(\hat{\boldsymbol{\phi}}_t./\mathbf{1}(\boldsymbol{D}(\hat{\boldsymbol{\phi}}_t)\hat{\boldsymbol{Q}}_t))\hat{\boldsymbol{Q}}_t$. And in the M-step, the transition matrix $\hat{\boldsymbol{Q}}_{t+1} = (\hat{q}_{t+1}(i,j))$ is updated by:

$$\hat{q}_{t+1}(i,j) = \frac{\hat{S}_{t+1}(i,j)}{\sum_{j=1}^{m}\hat{S}_{t+1}(i,j)}, \tag{12}$$

where $\hat{S}_{t+1}(i,j) = \sum_{k=1}^{m}\hat{\phi}_{t+1}(k)\hat{\rho}_{t+1}(i,j,k)$.

As can be seen, Eqs. (10) and (11) are the modifications of Eqs. (8) and (9) where $\hat{\boldsymbol{\phi}}_1, \hat{\boldsymbol{\phi}}_2, ..., \hat{\boldsymbol{\phi}}_t$ and $\hat{\rho}_1, \hat{\rho}_2, ..., \hat{\rho}_t$ did not depend on the parameter $\boldsymbol{Q}$, but rather $\hat{\boldsymbol{Q}}_t$, and thus do not need to be recalculated.

In the proposed on-line EM algorithm by Cappé (2011), a decreasing sequence of forgetting factors $\{\gamma_t\}_{t=1}^{\infty}$ is chosen such that $\sum_{t=1}^{\infty}\gamma_t = \infty$ and $\sum_{t=1}^{\infty}\gamma_t^2 < \infty$. The choice of γ_t strongly affects the convergence of the parameters. To converge to a stationary point one can choose $\gamma_t = 1/t^{\alpha}$ with $0.5 < \alpha < 1$, which is the common choice suggested in Cappé (2011). By setting γ_t to a fixed value, the algorithm will never converge to any fixed point but behave like a stochastic processes. As we will see later, this can be useful when the data comes from a non-stationary process, where the parameters are not fixed over time.

2.3.1 Setting forgetting factor

When using a fixed value for γ_t $(= \gamma)$ it is crucial that this value is well chosen. A smaller γ gives a more stable parameter trajectory, at the price of a slower adaptation. In the present form, it can be hard to see what a reasonable value of γ is. To show this more clearly, we introduce two explanatory parameters (K, R), which represent the weight, R, that is put on the K latest observations, when estimating $\boldsymbol{Q}$. So for instance, if $K = 100$, and $R = 0.9$, then the weight given to the hundred latest observations is such that, they represent 90% of the information from the data used to estimate the parameters.

To link the parameters K and R to γ, note that (11) is approximately a geometric series with ratio γ, thus approximately it holds that

$$\gamma\sum_{i=0}^{K}(1-\gamma)^i = R. \tag{13}$$

This gives an explicit γ for each (R, K).

A further issue is that in general, one observation does not contain equal information about all the entires in $\boldsymbol{Q}$, some states (events) might occur rarely and thus most observations contain no information about the corresponding column in the transition matrix. To address this, one can set a separate γ for each column. One way is to set $\gamma_{t,i} = \gamma \cdot (\boldsymbol{\pi}_t)_i$ where $\boldsymbol{\pi}_t$ is the averaged stationary distribution vector defined below.

2.4 On-line estimation of the number of events

In previous work, see Maghsood et al. (2015), the Viterbi algorithm introduced by Viterbi (1967) was used to calculate the number of driving events. However, the Viterbi algorithm requires access to the entire data sequences and thus can not be used for on-line estimation when the data is not stored. Instead we compute the expected number of events as follows.

Suppose that at each time t, the Markov chain $\{Z_t\}$ has transition matrix $\boldsymbol{Q_t}$. By solving the equation $(\boldsymbol{Q_t} - I)\boldsymbol{\pi_t} = \boldsymbol{0}$, one gets the stationary distribution of $\boldsymbol{Q}_t$. The expected number of i^{th} event for $\{Z_t\}_{t=0}^{T}$ is equivalent to the number of times that transitions $j \to i$ for all $j \neq i$ occur. In addition, one should consider the state at time zero, Z_0, which can also be i. Therefore, the expected number of i^{th} event up to time T is:

$$\eta_i(T) = E[I(Z_0 = i)] + E[\sum_{t=1}^{T} \xi_i(t)] = \pi_{0,i} + \sum_{t=1}^{T} \sum_{j \neq i} \pi_{t,j} q_t(j,i), \tag{14}$$

where $\xi_i(t) = \sum_{j \neq i} I(Z_{t-1} = j, Z_t = i)$.

The above formula works if we substitute $\boldsymbol{Q}_t$ with the on-line estimate $\hat{\boldsymbol{Q}}_t$ for each t. Then, one can compute and update the number of events based on each new observation.

2.5 HMMs with Laplace distribution

As mentioned in the introduction, we set the conditional distribution of Y_t given Z_t, denoted by $g_{\boldsymbol{\theta}}(i, y_t)$, to be a generalized asymmetric Laplace distribution (GAL), see Kotz et al. (2001). The GAL distribution is a flexible distribution with four parameters: $\boldsymbol{\delta}-$ location vector, $\boldsymbol{\mu}-$ shift vector, $\nu > 0-$ shape parameter, and $\boldsymbol{\Sigma}-$ scaling matrix and denoted by $GAL(\boldsymbol{\delta}, \boldsymbol{\mu}, \nu, \boldsymbol{\Sigma})$. The probability density function (pdf) of a $GAL(\boldsymbol{\delta}, \boldsymbol{\mu}, \nu, \boldsymbol{\Sigma})$ distribution is

$$\begin{aligned} g(\boldsymbol{y}) &= \frac{1}{\Gamma(1/\nu)\sqrt{2\pi}} \left(\frac{\sqrt{(\boldsymbol{y}-\boldsymbol{\delta})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}-\boldsymbol{\delta})}}{c_2} \right)^{\frac{1/\nu - d/2}{2}} e^{(\boldsymbol{y}-\boldsymbol{\delta})\Sigma^{-1}\boldsymbol{\mu}} \\ & \qquad K_{1/\nu - d/2}\left(c_2 \sqrt{(\boldsymbol{y}-\boldsymbol{\delta})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}-\boldsymbol{\delta})} \right), \end{aligned}$$

where d is the dimension of $\boldsymbol{Y}$, $c_2 = \sqrt{2 + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}$ and $K_{1/\nu - d/2}(.)$ is the modified Bessel function of the second kind. The normal mean variance mixture representation can give an intuitive feel of the distribution. That is a random variable $\boldsymbol{Y}$ having GAL distribution, and the following equality holds:

$$\boldsymbol{Y} \stackrel{d}{=} \boldsymbol{\delta} + \Gamma\boldsymbol{\mu} + \sqrt{\Gamma}\boldsymbol{\Sigma}^{1/2}\boldsymbol{Z},$$

where Γ is a Gamma distributed random variable with shape $1/\nu$ and scale one, and $\boldsymbol{Z}$ is a vector of d independent standard normal random variable. For more details see Barndorff-Nielsen et al. (1982).

3 Estimation of fatigue damage

Fatigue is a random process of material deterioration caused by variable stresses. For a vehicle, stresses depend on environmental loads, like road roughness, vehicle usage or driver's behavior.

Often, the rainflow cycles are calculated in order to describe the environmental loads, and the fatigue damage is then approximated by a function of the rainflow cycles.

Typically, the approximations are done in order to reduce the length of the load signals storing only the events relevant for fatigue. The reduced signal is then used to find the fatigue life of components in a laboratory (or to estimate the fatigue life mathematically). The reduction is mainly done in order to speed up the testing which is very expensive (or simplify calculations).

In this section, we present a method to approximate the environmental load using driving events. The method is similar to a well-known method in fatigue analysis, the rainflow filter method, see e.g. Johannesson and Speckert (2013). We show that one can explicitly calculate the expected damage intensity (which describes the expected life time of a component) on-line.

We start with a short introduction to rainflow cycles and expected damage, then show the approximation method that uses the driving event to derive the expected damage.

3.1 Rainflow counting distribution and the expected damage

The rainflow cycle count algorithm is one of the most commonly used methods to compute fatigue damage. The method was first proposed by Matsuishi and Endo (1968). Here, we use the definition given by Rychlik (1987) which is more suitable for a statistical analysis of damage. The rainflow cycles are defined as follows.

Assume that a load L_T, the processes up to time T, has N local maxima. Let M_i denote the height of the i^{th} local maximum. Denote by m_i^+ (m_i^-) the minimum value in forward (backward) direction from the location of M_i until L_T crosses M_i again. The rainflow minimum, m_i^{rfc}, is the maximum value of m_i^+ and m_i^-. The pair (m_i^{rfc}, M_i) is the i^{th} rainflow pair with the rainflow range $h_i(L_T) = M_i - m_i^{rfc}$. Figure 1 illustrates the definition of the rainflow cycles.

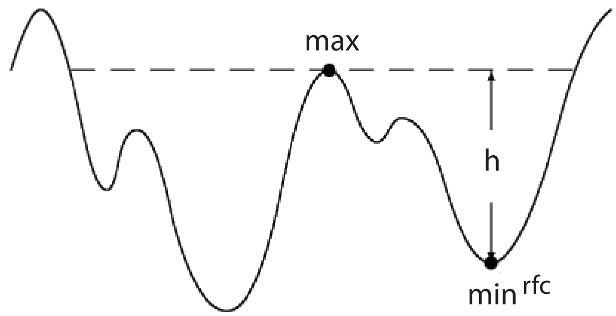


Figure 1: The rainflow cycle.

By using the rainflow cycles found in L_T, the fatigue damage can be defined by means of Palmgren-Miner (PM) rule, see Palmgren (1924), Miner (1945),

$$D_\beta(L_T) = \alpha \sum_{i=1}^{N} h_i(L_T)^\beta, \tag{15}$$

where α, β are material dependent constants. The parameter α^{-1} is equal to the predicted number of cycles with range one leading to fatigue failure (throughout the article it is assumed that α equals one). Various choices of the damage exponent β can be considered, like $\beta = 3$, which is the standard value for the crack growth process or $\beta = 5$ which is often used when a fatigue process is dominated by the crack initiation phase.

A more convenient representation, from computational viewpoint, of damage is:

$$D_\beta(L_T) = \beta(\beta-1)\int_{-\infty}^{+\infty}\int_{-\infty}^{v}(v-u)^{\beta-2}N^{osc}(u,v)\,du\,dv, \tag{16}$$

where $N^{osc}(u,v)$ is the number of interval ($[u,v]$) up-crossing by a load, see Rychlik (1993) for details.

Since L_T is a random process, one uses the expected damage as a tool to describe damage. The damage intensity of a process is

$$d_\beta = \lim_{T\to\infty}\frac{1}{T}E[D_\beta(L_T)]. \tag{17}$$

Finally, using Eq. (16), we get that

$$d_\beta = \beta(\beta-1)\int_{-\infty}^{+\infty}\int_{-\infty}^{v}(v-u)^{\beta-2}\mu^{osc}(u,v)\,du\,dv, \tag{18}$$

where

$$\mu^{osc}(u,v) = \lim_{T\to\infty}\frac{E\left[N^{osc}(u,v)\right]}{T}. \tag{19}$$

which is called the intensity of interval up-crossings.

3.2 Reduced load and expected damage given driving events

In general the lateral loads are not available and will vary between vehicles. The reduced load, we propose below, is constructed using estimated frequencies of driving events from the HMM, and the distributions of extreme loads associated with driving events, which can be measured on testing grounds or in laboratories.

We now describe how to construct a reduced load from the driving events left turn, LT, and right turn, RT (the method could of course be generalized to other driving events); these events are known to cause the majority of the damage for steering components. Let $\{Z_t\}_{t=0}^{T}$ be the hidden processes in a HMM, with three possible driving states right turn, left turn or straight forward, at time t. Define Z_i^* as the driving event representing the i^{th} turn, occurring in the time interval $[t_{i,start}, t_{i,stop}]$. We assume that Z_i^* equals one if the turn is left, and two if the turn is right. The relation between the two sequences $\{Z_i^*\}_{i=0}^{N}$ and $\{Z_t\}_{t=0}^{T}$ is that the event $\{Z_i^* = 1\}$(or $\{Z_i^* = 2\}$) is equivalent to that $Z_{t_{i,start}}, ..., Z_{t_{i,stop}}$ are all equal to, the same driving state, left turn (or right turn).

Now to create the reduced load, from the sequence driving events, assume that M_i and m_i are the i^{th} maximum and minimum load during a turn, that is

$$M_i = \max_{t \in I_i} L_t, \qquad m_i = \min_{t \in I_i} L_t, \tag{20}$$

where $I_i = [t_{i,start}, t_{i,stop}]$ represents the start and stop points of i^{th} turn. The reduced load $\{X_i\}_{i=0}^{2N}$ is defined as follows

$$X_i = \begin{cases} 0, & \text{if } i \text{ is odd integer,} \\ M_{i/2}, & \text{if } Z_{i/2}^* = 1, i \text{ is even integer,} \\ m_{i/2}, & \text{if } Z_{i/2}^* = 2, i \text{ is even integer.} \end{cases} \tag{21}$$

Here the zeros are inserted since between each left and right turn event there must be a straight forward event. Figure 2 illustrates a lateral load and the corresponding reduced load.

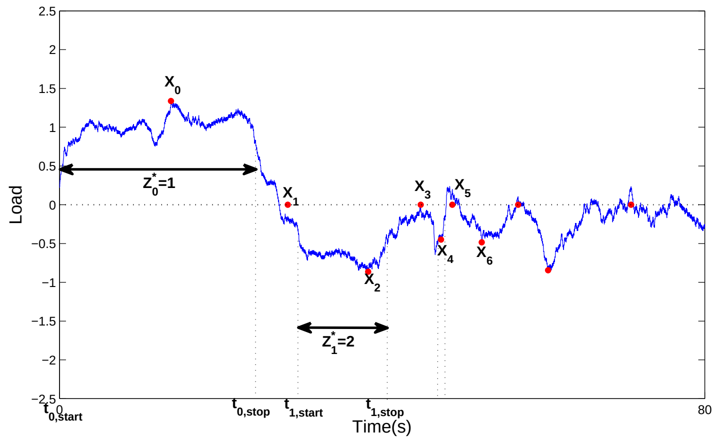


Figure 2: Reduced load represented by dots where the observed load is represented by the irregular solid line.

To compute the damage intensity d_β, per driving event, one needs the interval up-crossing intensity $\mu^{osc}(u,v)$ of $\{Z_i^*\}_{i=0}^N$. Assuming that both $\{M_i\}_{i=0}^N$ and $\{m_i\}_{i=0}^N$ are sequences of i.i.d. random variables, and that $\{Z_i^*\}_{i=0}^N$ is a Markov chain with transition matrix $\boldsymbol{P} = (p(k,j))$ (it can be derived from transition matrix $\boldsymbol{Q}$ in the HMM, see Appendix), one gets the closed form solution

$$\mu^{osc}(u,v) = \frac{1}{2}\begin{cases} \pi_2^* P(m_1 < u), & u < v < 0, \\ \pi_2^*\, P(m_1 < u)\, p_2(u,v), & u \le 0 \le v, \\ \pi_1^* P(M_1 > v), & 0 < u < v. \end{cases} \tag{22}$$

Here $\pi^* = (\pi_1^*, \pi_2^*)$ is the stationary distribution of the $\boldsymbol{P}$ and $p_2(u,v)$ can be derived from the equation system:

$$\begin{aligned} p_j(u,v) = & \, p(j,1)P(M_1 > v) + P(M_1 \le v)\, p(j,1)\, p_1(u,v) \\ & + P(m_1 \ge u)\, p(j,2)\, p_2(u,v),\; j = 1,2. \end{aligned} \tag{23}$$

For more details see Maghsood et al. (2015).

4 Examples

We evaluate the proposed algorithm with simulated and measured data sets. We consider the steering events occurring when the vehicle is driving at a speed higher than 10 km/h, e.g. when driving in curves. We estimate the number of left and right turns for a costumer. We further investigate the damage caused by steering events and compute the expected damage using the on-line estimation of transition matrix.

In our simulation study, a training set is used to estimate the parameters of the model which contains all steering events. We also use the simulation study to show the effects of different values of forgetting factor γ.

Finally, we use the measured data which is dedicated field measurements from a Volvo Truck. The measured signals come from the CAN (Controller Area Network) bus data, which is a systematic data acquisition and contains customer data.

4.1 Simulation study

We want to imitate a real journey during different road environments, such as city streets and highways. This is done by first generating a sequence of steering states using a Markov chain. We consider three states right turn (RT), left turn (LT) and straight forward (SF). We set these events as three hidden states and construct the HMM based on them as follows: We assume that the probabilities of going from a right turn to a left turn and vice versa are small and most often we will have straight forward after a right or a left turn. It has been also assumed that the average duration of straight forward during a city road is less than highway. Two different transition matrices $\boldsymbol{Q}_{city}$ and $\boldsymbol{Q}_{highway}$ have been considered for city and highway respectively:

$$\boldsymbol{Q}_{city} = \begin{array}{c c} & \begin{array}{ccc} \text{RT} & \text{SF} & \text{LT} \end{array} \\ \begin{array}{c} \text{RT} \\ \text{SF} \\ \text{LT} \end{array} & \begin{pmatrix} 0.85 & 0.1 & 0.05 \\ 0.025 & 0.95 & 0.025 \\ 0.05 & 0.1 & 0.85 \end{pmatrix} \end{array}, \boldsymbol{Q}_{highway} = \begin{array}{c c} & \begin{array}{ccc} \text{RT} & \text{SF} & \text{LT} \end{array} \\ \begin{array}{c} \text{RT} \\ \text{SF} \\ \text{LT} \end{array} & \begin{pmatrix} 0.90 & 0.08 & 0.02 \\ 0.005 & 0.99 & 0.005 \\ 0.02 & 0.08 & 0.90 \end{pmatrix} \end{array}.$$

Second, we use Laplace distribution to simulate the lateral acceleration signal, Y_t. The Laplace parameters $(\delta, \mu, \nu, \Sigma)$ for each state are set as follows:

- $\delta_{RT} = -\delta_{LT} = -1,\ \delta_{SF} = 0,$
- $\mu_{RT} = -\mu_{LT} = -0.5,\ \mu_{SF} = 0,$
- $\nu_{RT} = \nu_{LT} = 10,\ \nu_{SF} = 0.5,$
- $\Sigma_{RT} = \Sigma_{LT} = 0.2,\ \Sigma_{SF} = 1.$

The fitted distributions for lateral acceleration values within each state are shown in Figure 3.

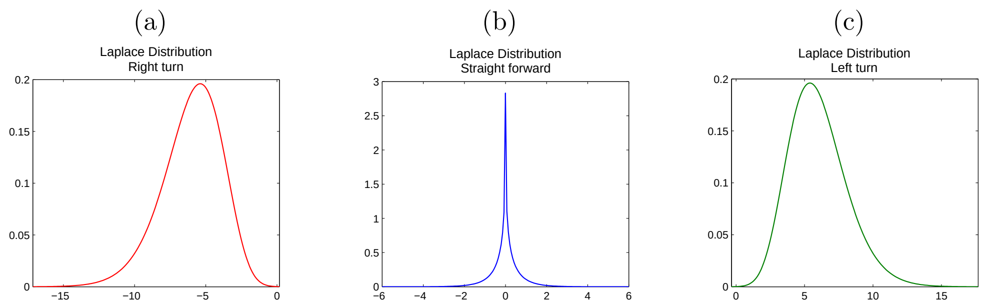


Figure 3: (a), (b) and (c) represent the Laplace distributions fitted on lateral acceleration values for right turns, straight forward and left turns respectively.

We compare four different values of γ_t for the estimation of the transition matrix. First, we set $\gamma_t = 1/t^{\alpha}$ where $\alpha = 0.9$. This value of forgetting factor satisfies the convergence conditions given by Cappé (2011). Second we use three different values of fixed γ, 0.01, 0.002 and 0.001–corresponding to $R = 0.9$ and $K = 200, 1000$ and 2400 (which corresponds to a duration 2 min, 10 min, and 20 min) in Eq. (13). Figure 4 shows the estimated diagonal elements of the transition matrices for one simulated signal. The simulated signal represents a journey on a city road, a highway and then back to a city road and again highway over 10^5 seconds, where the sampling period is 1/2 seconds. The straight thick black lines show the diagonal elements of true transition matrices $\boldsymbol{Q}_{city}$ and $\boldsymbol{Q}_{highway}$.

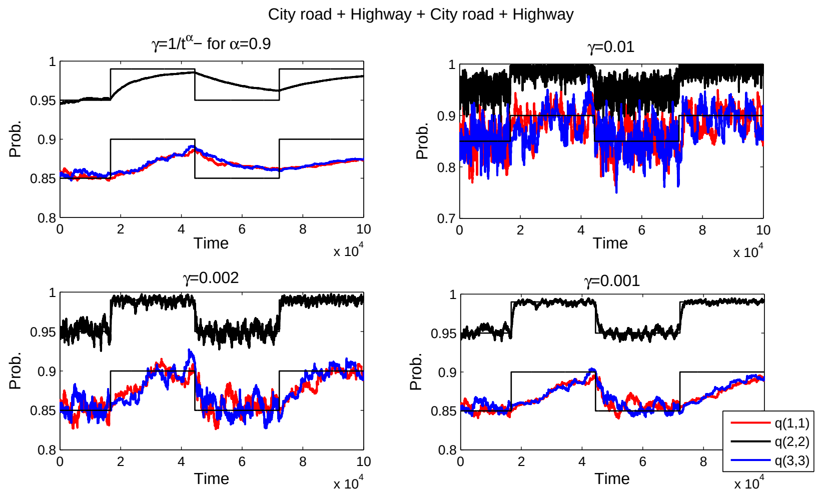


Figure 4: Diagonal elements of on-line estimated transition matrix, simulated signal from City road+Highway+City road+Highway, with four different values of γ. Straight thick black lines show the diagonal elements of true transition matrices $\boldsymbol{Q}_{city}$ and $\boldsymbol{Q}_{highway}$.

In Figure 4, one can see that the on-line algorithm with variable γ can not follow the changes of the parameters well and that the adaption diminishes over time, as is to be expected. The fixed forgetting factor, however, seems to adapt well to the chaining environment.

Expected number of events

Here, we compute the expected number of turns. We simulate independently hundred signals in order to investigate the accuracy of the on-line algorithm with different forgetting factors γ. In that case, we choose as before four different values for the forgetting factor. The values of the fixed forgetting factors correspond to $R = 0.9$ and $K = 200, 1000$ and 2400 in Eq. (13).

We perform 100 simulations and estimate the intensities of occurrences of turns by Eq. (14):

$$\eta_{LT} = \pi_{0,3} + \sum_{t=1}^{T} (\pi_{t,2}\hat{q}_t(2,3) + \pi_{t,1}\hat{q}_t(1,3)), \tag{24}$$

$$\eta_{RT} = \pi_{0,1} + \sum_{t=1}^{T} (\pi_{t,2}\hat{q}_t(2,1) + \pi_{t,3}\hat{q}_t(3,1)). \tag{25}$$

In order to validate the results, we compute an error rate which is the difference between the estimated and observed number of turns in each simulation. The expected number of turns from the model (using $\boldsymbol{Q}_{city}$ and $\boldsymbol{Q}_{highway}$) are $\eta_{LT} =$

$\eta_{RT} = 2840$. The average number of observed left and right turns are $n_{LT} = 2834$ and $n_{RT} = 2836$, respectively. The average and the standard deviations of errors for 100 simulations are computed. The results are presented in Table 1. According to the average error, the forgetting factor $\gamma_t = 0.002$ performs the best. However there is, surprisingly, only a small difference between all the fixed forgetting factors.

Table 1: The expected number of turns estimated by on-line algorithm and Eqs. (24), (25). The errors are the average of the differences between the estimated and observed number of turns.

On-line algorithm								
γ_t	$1/t^{0.9}$		0.01		0.002		0.001	
Turns	η_{LT}	η_{RT}	η_{LT}	η_{RT}	η_{LT}	η_{RT}	η_{LT}	η_{RT}
Mean Est.	3236	3241	2928	2932	2882	2886	2920	2924
Mean Error	402.48	405.30	94.40	96.46	48.45	49.93	86.84	88.68
Std Error	28.45	33.78	15.41	15.79	16.61	17.77	20.65	21.43

In our previous work, an HMM combined with a Viterbi algorithm, see Viterbi (1967), has been used to identify the driving events. The Viterbi algorithm gives a reconstructed sequence of events which maximizes the conditional probability of the observation sequence. In that approach, all data has to be used to estimate the driving events and is thus not suitable to on-board usage in a vehicle. However, in order to compare the previously proposed approach with the on-line estimation and to evaluate the frequencies of driving events, we also compute the number of turns by the Viterbi algorithm for each simulation. The counted number of turns from the Viterbi algorithm are on average $\eta_{LT} = 2923$ and $\eta_{RT} = 2925$. One can see that the Viterbi algorithm overestimates the number of turns.

Damage investigation

In this section we compute the damage intensity per kilometer based on on-line estimation of transition matrix. We use one of the simulated lateral acceleration signal in order to calculate the damage. The speed of the vehicle is considered 50 kilometers per hour and the mileage is 1000 km (for a sampling period of 1/2 seconds). We split the signal into 1000 equally sized frames. For each frame, the expected number of turns are computed by $\Delta\eta_k = \eta_k - \eta_{k-1}$ where η_k is the estimated number of turns occurring over the first k kilometers. The expected damage based on turns for each frame is calculated by:

$$\Delta d_k = \Delta\eta_k d_k,$$

where d_k is the expected damage per turn and calculated by means of Eqs. (18) and (22). The empirical distribution of M_i and m_i are used to calculate the intensity of interval crossings $\mu^{osc}(u, v)$. We use the on-line estimation of transition matrix $\boldsymbol{Q}$ with $\gamma = 0.002$ to estimate the transition matrix $\boldsymbol{P}$ by using Eqs. (.28) and (.27), see Appendix. The result for damage exponent $\beta = 3$ is shown in Figure 5. The straight thick red line shows $\Delta d_k(\boldsymbol{Q}_{true})$, which is the damage intensity computed using the model transition matrices $\boldsymbol{Q}_{city}$ and $\boldsymbol{Q}_{highway}$ for city and highway respectively. We can observe the change in damage between highway and city road. As might be expected the damage intensities (per km) estimated for the city are higher than for highway, since the number of turns occurring in a city road are larger than on a highway.

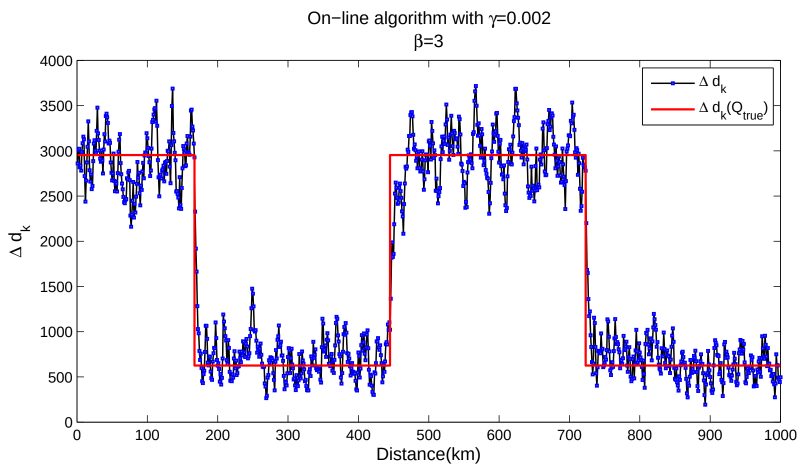


Figure 5: Damage intensity per km according to the on-line estimation of transition matrix with $\gamma = 0.002$. The upper plot shows the results for damage exponent $\beta = 3$. The straight thick red line shows $\Delta d_k(\boldsymbol{Q}_{true})$ which is the damage intensity computed using model transition matrices $\boldsymbol{Q}_{city}$ and $\boldsymbol{Q}_{highway}$ for city and highway, respectively.

Further, the expected damage from the model (theoretical damage) is compared with the total damage and the damage calculated from the reduced load. One can see that the expected damage for the whole signal – based on on-line estimation of transition matrix– is equal to $\sum_{k=1}^{1000} \Delta d_k$. The total damage is calculated from the lateral acceleration signal using the rainflow method. The damage evaluated for the load (lateral acceleration), reduced load and the expected damage is compared in Table 2. The numerical integration in (18) as well as the rainflow cycle counting has been done using the WAFO (Wave Analysis for Fatigue and Oceanography) toolbox, see Brodtkorb et al. (2000); WAFO Group (2011a,b).

Table 2: Comparison of damage computed for the simulated load, the corresponding reduced load and the expected damage.

Damage	Total	Reduced load	Expected On-line with $\gamma = 0.002$
$\beta = 3$	$1.88 \cdot 10^6$	$1.68 \cdot 10^6$	$1.68 \cdot 10^6$
$\beta = 5$	$1.77 \cdot 10^8$	$1.72 \cdot 10^8$	$1.67 \cdot 10^8$

Figure 5 and Table 2 demonstrate high accuracy of the proposed approach to estimate the expected damage for the studied load. Obviously this load is a realistic mathematical model of a real load. In the next section we will apply our method to estimate the steering events and compute the damage for a measured load on a VOLVO truck.

4.2 On-board logging data from Volvo

To evaluate the method on a real data set, we study field measurements coming from a Volvo Truck. We use the measured lateral acceleration signal from the CAN (Controller Area Network) bus data.

We fit the Laplace distribution for the lateral acceleration within each steering state. To estimate the Laplace distribution parameters considered, we need a training set which contains all history about the curves. We detect the events manually by looking at video recordings from the truck cabin to see what happened during the driving. The manual detections are not completely correct because of the visual errors and the low quality of videos used for the manual detection.

The on-line algorithm is used to count the number of left and right turns. Figure 6 shows the estimation results using on-line algorithm with $\gamma_t = 0.0008$, $(R = 0.8, K = 2000)$ for the measured signal. It is interesting to note that there is a sudden change in the driving environment after around 5000 sec.

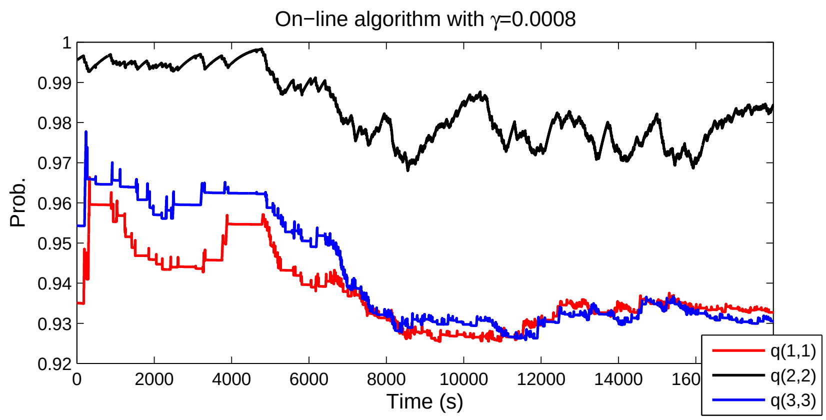


Figure 6: Diagonal elements from on-line estimation of transition matrix with $\gamma_t = 0.0008$ for measured data

The expected number of left and right turns computed by on-line algorithm are $\eta_{LT} = 228$ and $\eta_{RT} = 241$ respectively.

Damage investigation

Here, we compute the damage intensity based on the model. In order to do that we split data into the frames containing 250 seconds (approximately 4-5 km) of measurement and we compute the distance based on the average speed in each frame. Figure 7 shows the expected damage based on turns computed by $\Delta d_k = \Delta\eta_k d_k$ where $\Delta\eta_k = \eta_k - \eta_{k-1}$ and n_k is the estimated number of turns occurring over the first k kilometers. Here, the results are based on the damage exponent $\beta = 3$.

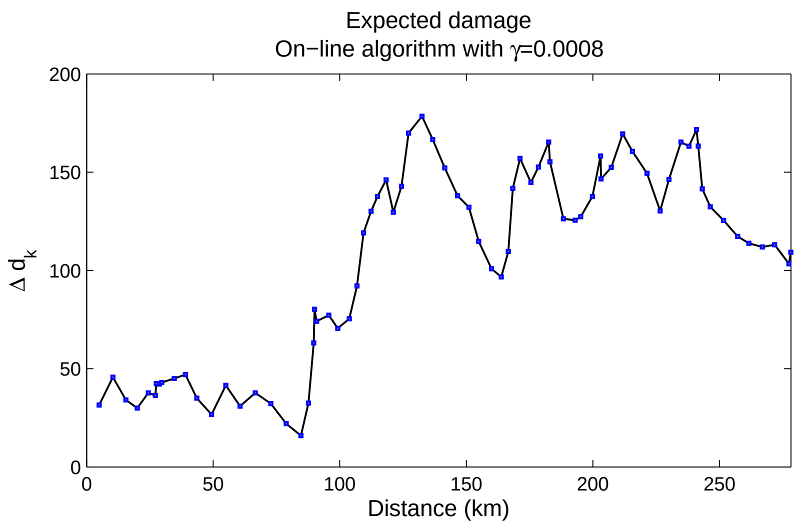


Figure 7: Damage intensity with damage exponent $\beta = 3$ regarding mileage. The on-line estimation of transition matrix with $\gamma = 0.0008$ has been used to estimate the expected damage.

The total expected damage using on-line estimation of transition matrix can be computed by $\sum_{k=1} \Delta d_k$. The damage evaluated for the load (lateral acceleration), reduced load and the expected damage is compared in Table 3. The Rayleigh distributions which have been fitted to positive and negative values of the reduced load are

$$P(M_1 > v) = e^{-\frac{1}{2}\left(\frac{v}{2.2}\right)^2},\ v \geq 0, \qquad P(m_1 < u) = e^{-\frac{1}{2}\left(\frac{u}{2.3}\right)^2},\ u \leq 0.$$

Table 3: Comparison of damage values computed from the measured load, the corresponding reduced load and the expected damage.

Damage	Total	Reduced load	Expected On-line with $\gamma = 0.0008$
$\beta = 3$	$8.1 \cdot 10^3$	$7.4 \cdot 10^3$	$7.7 \cdot 10^3$
$\beta = 5$	$1.5 \cdot 10^5$	$1.5 \cdot 10^5$	$1.9 \cdot 10^5$

We also compare the damage accumulation process from the model, $\sum_{k=1} \Delta d_k$, with the empirical accumulated damage in the signal. The expected damage based on fitted model will be called the theoretical damage. Figure 8 shows the theoretical and observed accumulated damage processes. It can be seen that the accumulated damage from the model is close to the observed damage and there are two damage rates in both theoretical and observed damage processes.

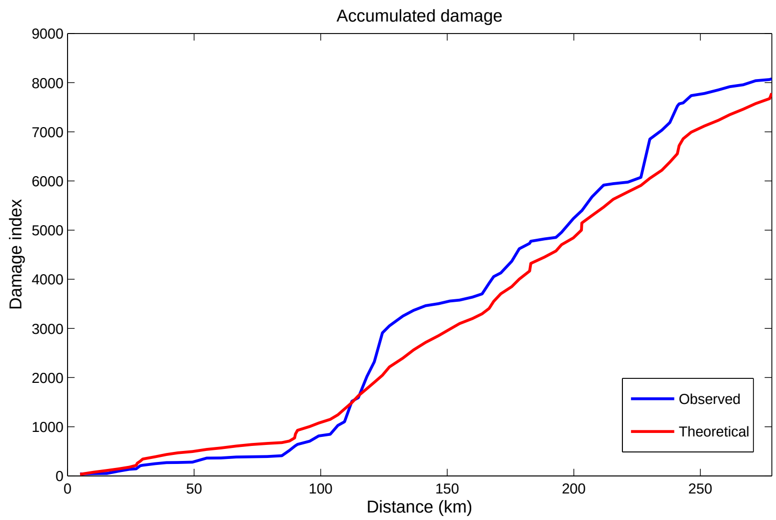


Figure 8: The theoretical and observed accumulated damage processes for damage exponent $\beta = 3$. The on-line estimation of transition matrix with $\gamma = 0.0008$ has been used to estimate the expected damage.

Results shown in Figure 8 and Table 3 demonstrate the accuracy of the proposed methodology for this measured load.

5 Conclusion

In this article, we have derived a method to estimate the number of driving events for a vehicle using the CAN data through the use of an HMM. The method uses an on-line EM algorithm to estimate the parameters of the HMM. The on-line version has three major advantages over the regular EM algorithm, making it possible to implement the method on-board a vehicle: the computational complexity of each iteration of the algorithm is $\mathcal{O}(1)$, making it a computationally tractable method; the parameters are estimated without the need to store any data; the formulation of the on-line algorithm allows for an adaptive parameter estimation method, using a fixed forgetting factor, so that the parameters can adapt over chaining driving environment.

The proposed estimation algorithm was validated using simulated and measured data sets. The results show that the on-line algorithm works well and can adapt to a chaining environment when the driving conditions are not constant over time.

Acknowledgment

We are thankful to Prof. Igor Rychlik and Dr. Pär Johannesson for their 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. The second author has been supported by the Knut and Alice Wallenberg foundation.

References

Arvastson, L., Olsson, H., and Holst, J. (2000). Asymptotic bias in parameter estimation of ar-processes using recursive least squares with exponential forgetting. *Scandinavian Journal of Statistics*, 27(1):177–192.

Barndorff-Nielsen, O., Kent, J., and Sorensen, M. (1982). Normal variance-mean mixtures and z distributions. *International Statistical Review*, 50:145–159.

Berndt, H. and Dietmayer, K. (2009). Driver intention inference with vehicle onboard sensors. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 102–107, Pune.

Bogsjö, K., Podgorski, K., and Rychlik, I. (2012). Models for road surface roughness. *Vehicle System Dynamics*, 50:725–747.

Brodtkorb, P. A., Johannesson, P., Lindgren, G., Rychlik, I., Rydén, J., and Sjö, E. (2000). WAFO – a Matlab toolbox for analysis of random waves and loads. In *Proceedings of the 10th International Offshore and Polar Engineering conference, Seattle*, volume III, pages 343–350.

Cappé, O. (2011). Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*, 20:3:728–749.

Cappé, O., Moulines, E., and Rydén, T., editors (2005). *Inference in Hidden Markov Models*. Springer.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer.

Johannesson, P. and Speckert, M., editors (2013). *Guide to Load Analysis for Durability in Vehicle Engineering*. Wiley:Chichester.

Kotz, S., Kozubowski, T., and Podgorski, K. (2001). *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media.

Kvanström, M., Podgórski, K., and Rychlik., I. (2013). Laplace moving average model for multi-axial responses in fatigue analysis of a cultivator. *Probabilistic Engineering Mechanics*, 34:12–25.

Lennart, L. and Söderström, T. (1983). Theory and practice of recursive identification.

Maghsood, R. and Johannesson, P. (2013). Detection of the curves based on lateral acceleration using hidden Markov models. *Procedia Engineering*, 66:425–434.

Maghsood, R. and Johannesson, P. (2016). Detection of steering events based on vehicle logging data using hidden Markov models. *International Journal of Vehicle Design*, 70(3):278 295.

Maghsood, R., Rychlik, I., and Wallin, J. (2015). Modeling extreme loads acting on steering components using driving events. *Probabilistic Engineering Mechanics*, 41:13–20.

Matsuishi, M. and Endo, T. (1968). Fatigue of metals subjected to varying stress. *Japan Society of Mechanical Engineers. In Japanese.*

Miner, M. A. (1945). Cumulative damage in fatigue. *Journal of Applied Mechanics*, 12:A159–A164.

Mitrović, D. (2004). *Learning Driving Patterns to Support Navigation.* PhD thesis, University of Canterbury, New Zealand.

Mitrović, D. (2005). Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):198–205.

Palmgren, A. (1924). Die Lebensdauer von Kugellagern. *Zeitschrift des Vereins Deutscher Ingenieure*, 68:339–341. *In German.*

Rychlik, I. (1987). A new definition of the rainflow cycle counting method. *International Journal of Fatigue*, 9:119–121.

Rychlik, I. (1993). Note on cycle counts in irregular loads. *Fatigue & Fracture of Engineering Materials & Structures*, 16:377–390.

Vahidi, A., Stefanopoulou, A., and Peng, H. (2005). Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics*, 43(1):31–55.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13(2):260–269.

WAFO Group (2011a). WAFO – a Matlab toolbox for analysis of random waves and loads, tutorial for WAFO 2.5. Mathematical Statistics, Lund University.

WAFO Group (2011b). WAFO – a Matlab Toolbox for Analysis of Random Waves and Loads, Version 2.5, 07-Feb-2011. Mathematical Statistics, Lund University.
Web: `http://www.maths.lth.se/matstat/wafo/` (Accessed 24 January 2014).

Zeitouni, O. and Dembo, A. (1988). Exact filters for the estimation of the number of transitions of finite-state continuous-time Markov processes. *IEEE Transactions on Information Theory*, 34(4):890–893.

Appendix

Derivation of transition matrix of driving events

To construct the sequence $\{Z_i^*\}_{i=0}^N$, of driving events, let $\{t_k\}_{k=0}^N$ be the indices in $\{t : Z_{t_k} \neq Z_{t_k-1} \cap Z_{t_k} \neq SF\}$, then

$$Z_k^* = \begin{cases} 1, & \text{if } Z_{t_k} = \text{LT}, \\ 2, & \text{if } Z_{t_k} = \text{RT}. \end{cases} \tag{.26}$$

Assume that Z^* has transition matrix $\boldsymbol{P} = (p(k,j))$. Note that the hidden process $\{Z_t\}_{t=0}^T$ in HMM has three states "1" = RT, "2" = SF and "3" = LT. One can now derive the transition matrix $\boldsymbol{P}$ from the transition matrix of the HMM $\hat{\boldsymbol{Q}}$ as follows:

$$\hat{p}(1,1) = \frac{\hat{q}(3,2)\hat{q}(2,3)}{(1-\hat{q}(2,2))(1-\hat{q}(3,3))}, \tag{.27}$$

$$\hat{p}(2,2) = \frac{\hat{q}(1,2)\hat{q}(2,1)}{(1-\hat{q}(2,2))(1-\hat{q}(1,1))}. \tag{.28}$$

As proof, we consider for instance the probability of going from LT to RT in Z_i^* which can be computed as follows:

$$\begin{aligned}
\hat{p}(1,2) &= P(Z_{i+1}^* = 2 | Z_i^* = 1) \\
&= P(Z_{t_{i,start}:t_{i,stop}} = 3, Z_{t_{i,stop}+1:t_{i+1,start}-1} = 2, Z_{t_{i+1,start}:t_{i+1,stop}} = 1) \\
&\qquad\qquad + P(Z_{t_{i,start}:t_{i,stop}} = 3, Z_{t_{i+1,start}:t_{i+1,stop}} = 1), \\
&= \big(\hat{q}(3,2)\left(1 + \hat{q}(2,2) + \hat{q}^2(2,2) + \ldots\right)\hat{q}(2,1) + \hat{q}(3,1)\big) \\
&\qquad\qquad (1 + \hat{q}(3,3) + \hat{q}^2(3,3) + \ldots), \\
&= \left[\frac{\hat{q}(3,2)\hat{q}(2,1)}{1-\hat{q}(2,2)} + \hat{q}(3,1)\right]\frac{1}{1-\hat{q}(3,3)},
\end{aligned}$$

where $Z_{t_{i,start}:t_{i,stop}}$ represents the sequence of consecutive driving states $Z_{t_{i,start}}$, ..., $Z_{t_{i,stop}}$.