Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2021

Torque-based Load Estimation for Passenger Vehicles

Tobias Nyberg

LINKÖPING UNIVERSITY

Master of Science Thesis in Electrical Engineering

Torque-based Load Estimation for Passenger Vehicles

Tobias Nyberg

LiTH-ISY-EX--21/5436--SE

Supervisor: **Daniel Bossér**
ISY, Linköpings universitet
Jonah Ekelund
Syntronic AB

Examiner: **Gustaf Hendeby**
ISY, Linköpings universitet

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

An accurate estimate of the mass of a passenger vehicle is important for several safety systems and environmental aspects. In this thesis, an algorithm for estimating the mass of a passenger vehicle using the recursive least squares method is presented. The algorithm is based on a physical model of the vehicle and is designed to be able to run in real-time onboard a vehicle and uses the wheel torque signal calculated in the electrical control unit in the engine. Therefore no estimation of the powertrain is needed. This is one contribution that distinguishes this thesis from previous work on the same topic, which has used the engine torque. The benefit of this is that no estimation of the dynamics in the powertrain is needed. The drawback of using this method is that the algorithm is dependent on the accuracy of the estimation done in the engine electrical control unit.

Two different versions of the recursive least squares method (RLS) have been developed - one with a single forgetting factor and one with two forgetting factors.

The estimation performance of the two versions are compared on several different real-world driving scenarios, which include driving on country roads, highways, and city roads, and different loads in the vehicle. The algorithm with a single forgetting factor estimates the mass with an average error for all tests of 4.42 % and the algorithm with multiple forgetting factors estimates the mass with an average error of 4.15 %, which is in line with state-of-the-art algorithms that are presented in other studies.

In a sensitivity analysis, it is shown that the algorithms are robust to changes in the drag coefficient. The single forgetting factor algorithm is robust to changes in the rolling resistance coefficient whereas the multiple forgetting factor algorithm needs the rolling resistance coefficient to be estimated with fairly good accuracy. Both versions of the algorithm need to know the wheel radius with an accuracy of 90 %.

The results show that the algorithms estimate the mass accurately for all three different driving scenarios and estimate highway roads best with an average error of 2.83 % and 2.69 % for the single forgetting factor algorithm and the multiple forgetting factor algorithm, respectively. The results indicate it is possible to use either algorithm in a real-world scenario, where the choice of which algorithm depends on sought-after robustness.

Acknowledgments

I would like to thank Syntronic for giving me the opportunity to do this master thesis. I want to especially thank my supervisor at Syntronic, Jonah Ekelund, for all his help and input during this project. His help was crucial to make this thesis possible. I also want to thank my supervisor Daniel Bossér and my examiner Gustaf Hendeby for all their input and knowledge which contributed greatly to this thesis.

Linköping, August 2021
Tobias Nyberg

Contents

1 Introduction

The aim of this thesis is to develop an algorithm that estimates the mass of a passenger vehicle. This thesis project was carried out at Syntronic AB. In this chapter, a background to the problem, related work, and a problem formulation is presented. Some limitations to the thesis are also presented.

1.1 Background

Detecting overload and accurately determining the mass of a passenger vehicle is important from a security and environmental point of view. The braking distance is increased with the increased weight of a vehicle and increased weight changes the vehicle dynamics. Accurately estimating the mass is important for the anti-lock brake controller and can also help reduce emissions. By knowing the mass the appropriate gear can be selected, the cruise control can be more economic which both helps reduce emissions and improve the comfort of the ride.

Given the vehicle dynamics equation, the vehicle mass and road grade needs to be estimated simultaneously on-board. There exist some hardware-based solutions, which have the need for extra sensors in the cars, which is costly to put in all production cars. By using the signals that already are available on all vehicles and no extra sensors a software-based and cost-effective solution can be developed to estimate the mass of the vehicle.

The aim of this thesis is to develop an algorithm that estimates the mass of a passenger vehicle only using signals already available on the CAN-bus.

1.2 Related work

There have been several studies that have been studying load estimation using different techniques. Ghosh et al. [6] estimate the mass and road grade by using a recursive least squares, RLS, estimator combined with a torque observer that estimates the wheel torque. The authors also filter the signals to get rid of high-frequency factors such as drag and rolling resistance. The results show that the algorithm estimates the mass within 5 % of the actual mass and when using the torque observer it converges to the actual value much faster than without the torque observer. By filtering the signals their algorithm is not sensitive to the effects of drag and rolling resistance.

In [20] the authors use an observer-based parameter estimator and then use two different methods for smoothing the mass estimate. They use a Kalman filter and an analysis of the mass parameter to determine periods of convergence. They achieve a mass estimate within 10 % and the estimate settles within 60 seconds.

Lin et al. [11] also use an RLS algorithm to estimate the mass and discuss the impact of system errors in the model. They treat the system error as a parameter to estimate in the RLS instead of assuming it to be Gaussian. They discuss several parameters that can have an impact on estimates, e.g. accuracy of signals, tire slip, parameters that change during the motion, such as tire pressure. When not considering the system error the mass estimate error is 16 % and when considering it the error is 9 %. The algorithm was used on a heavy-duty vehicle.

Rezaeian et al. [14] estimate the forces acting on the tires of a passenger vehicle. To estimate the forces a mass estimate is needed. The authors do the mass estimation by using a longitudinal vehicle model and an RLS algorithm to estimate the mass. They achieve an accuracy of the mass estimate of 3 %, but in the estimation, they have assumed the center of gravity to be known and yaw rate to be measured and this may not be available in all cars.

Vahidi et al. [16] estimate the road grade and the mass of a heavy-duty vehicle (HDV) using two different methods. First, they estimate the mass and road grade with an observer and then with an RLS algorithm. They also investigate the RLS algorithm with two different forgetting factors, one for mass and one for road grade. When using a single forgetting factor the estimates are very poor and only converge to the actual values when the road grade is constant. Using multiple forgetting factors they show that both the estimates can become very accurate. This article is the basis of [17] where the authors use this theory to conduct real-life experiments.

Vahidi et al. [18] expand this theory by proposing an ad hoc modification of the update law for the gain in the RLS. The authors also investigate the performance of the algorithm on experimental data and discuss how the noise in the signals affects the estimation. They show that during times without gearshift the algorithm estimates the mass with a maximum error of 3 %. During gearshifts, the estimate overshoots, and therefore they propose a solution that turns off the estimation during those periods. This approach has been proven to be very slow and inaccurate when applied to passenger vehicles [5].

In [5] the authors propose a new algorithm that is based on that the inertial dynamics dominate vehicle motion over certain types of maneuvers. During those maneuvers, the mass estimation is done with an RLS algorithm. Together with band-pass filtering to get rid of high-frequency components the algorithm converges to an estimate near the actual mass fast. No values of accuracy are presented. In the article, they discuss that the algorithm is dependent on the persistence of excitation which may be a problem if the vehicle is driven carefully.

Jonsson Holm [8] estimates the mass of an HDV and the road grade using an extended Kalman filter with a longitudinal model. Two different methods are used, one where both the mass and the road grade are estimated and one where the road grade is measured with a sensor and only the mass is estimated. Jonsson Holm is showing that when estimating both the mass and the road grade the mass estimate is in the range of 5 % of the actual value, but when using the measurement of the road grade the mass estimate is within 2 % of the actual value.

In [9] the authors investigate four different methods that estimate mass and/or road grade. These include recursive least squares (RLS) with multiple forgetting factors, extended Kalman filtering, a dynamic grade observer, and a method developed in this research that is parallel mass and grade estimation using a longitudinal accelerometer. It is concluded that the RLS and EKF give an estimate within 5 % of their actual value when given a good initial guess.

Kim et al. [10] propose a combined longitudinal dynamics and roll dynamics to be able to estimate the mass during all types of driving. They use a model for roll dynamics during cornering and have two different RLS-algorithms where one algorithm is being run when moving longitudinal and one during cornering. The two different algorithms are then fused with a weighting function to give a good estimate. This is verified in simulations and they get good results. However, in the roll dynamics, they assume the center of gravity is known, which can vary with different loads and can be difficult to measure. They have only verified the results in simulation and not in real-world experiments.

To summarize most of the previous work has been done on heavy-duty vehicles and all of the previous work has used engine torque and estimated the driveline dynamics to acquire wheel force. Most of the literature on passenger vehicles has used additional sensors or highlighted the importance of accurately estimating the power train inertia. In most of the literature, only one driving scenario has been investigated and the error of the estimated mass is around 5-10 %.

1.3 Problem formulation

The purpose of this thesis is to develop an algorithm able to estimate the mass of a vehicle. This thesis will focus on answering the following questions:

1. How to adapt previously developed algorithms to incorporate wheel torque instead of engine torque?

2. How accurate is the algorithm compared to other algorithms that have been developed in terms of root mean square error and mean error?
3. Which driving scenarios have the largest effect on the performance of the algorithm?
4. Which parameters have the largest effect on the algorithm?

1.4 Limitations

Due to the scope of this thesis, the vehicle is not equipped with any additional sensors such as inclination sensors. This thesis only focuses on mass estimation, and therefore the road grade estimation is not of interest. As a consequence, no ground truth for road grade was calculated. When examining the results of the algorithm, only the result of the mass estimate is of interest.

1.5 Outline

The outline of this thesis is as follows. Chapter 2 provides a model for the vehicle dynamics and a mathematical description of the RLS method. It also presents the evaluation metrics that will be used throughout the thesis. Chapter 3 the proposed algorithm is presented. This includes how the vehicle model is incorporated in the RLS method, how the resampling and signal filtering is done. It also includes a description of a longitudinal motion detector that turns on the estimator when several conditions are met. This is used to conclude when the vehicle model is not valid and pause the estimation during these times. In Chapter 4 the experiments are presented and the results are presented and discussed. This also includes a sensitivity analysis. Finally, Chapter 5 provides conclusions to the problem formulation and also provides some remarks on future work.

2 Theory

This chapter presents the vehicle dynamics and a model for a longitudinal motion. It also provides a mathematical background for the RLS method, which is a common method in these applications and will be used in the estimator algorithm. Both the standard method and one with forgetting factors will be presented. The evaluation metrics that will be used are also presented.

2.1 Vehicle longitudinal dynamics

Newton's second law of motion states that the relation between force and acceleration is

$$F = ma, \tag{2.1}$$

where m is the mass of the object and a is the acceleration. The main forces acting on a vehicle when traveling in a longitudinal motion are shown in Figure 2.1. The forces acting on a vehicle combined with (2.1) gives an equation for the longitudinal motion as

$$ma = F_{\text{wheel}} - F_{\text{brake}} - F_{\text{drag}} - F_{\text{rollresist}} - F_{\text{roadgrade}}. \tag{2.2}$$

The traction force, F_{wheel}, depends on the engine torque and dynamics in the transmission. The brake force, F_{brake}, depends on the brake pressure. F_{drag} is the aerodynamic drag which depends on the vehicle longitudinal velocity v, the effective frontal surface of the vehicle S, the air density ρ and the aerodynamic drag coefficient C_d as

$$F_{\text{drag}} = \frac{1}{2}\rho S C_d v^2. \tag{2.3}$$

The rolling resistance $F_{\text{rollresist}}$ and resistance force $F_{\text{roadgrade}}$ are

$$F_{\text{rollresist}} = mgC_r \cos\theta \tag{2.4}$$

$$F_{\text{roadgrade}} = mg\sin\theta, \tag{2.5}$$

where m is the mass of the vehicle, g is the gravitational acceleration, C_r is the rolling resistance coefficient and θ is the road grade angle [6].

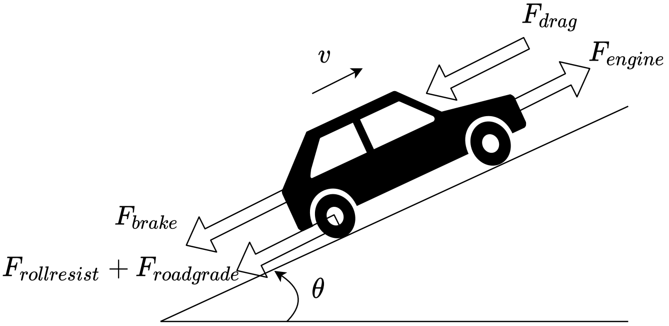


Figure 2.1: *Free body diagram of a vehicle travelling in a longitudinal motion.*

We can write (2.2) as

$$a = (F_{\text{wheel}} - F_{\text{brake}} - \frac{1}{2}\rho S C_d v^2)\frac{1}{m} - g(C_r\cos\theta + \sin\theta), \tag{2.6}$$

and by using the substitution $\theta_\mu = \arctan C_r$ and $a = \dot{v}$ we get

$$\dot{v} = (F_{\text{wheel}} - F_{\text{brake}} - \frac{1}{2}\rho S C_d v^2)\frac{1}{m} - \frac{g}{\cos\theta_\mu}(\sin(\theta + \theta_\mu)), \tag{2.7}$$

which can be written as

$$\dot{v} = \begin{bmatrix}\phi_1 & \phi_2\end{bmatrix}\begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}, \tag{2.8}$$

where

$$\begin{aligned}
\phi_1 &= F_{\text{wheel}} - F_{\text{brake}} - \frac{1}{2}\rho S C_d v^2 && \text{(2.9a)}\\
\phi_2 &= -\frac{g}{\cos\theta_\mu} && \text{(2.9b)}\\
\theta_1 &= \frac{1}{m} && \text{(2.9c)}\\
\theta_2 &= \sin(\theta + \theta_\mu). && \text{(2.9d)}
\end{aligned}$$

The unknown parameters are

$$\theta = \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix} \tag{2.10}$$

and

$$\dot{v}_x, \;\; \phi = \begin{bmatrix}\phi_1 \\ \phi_2\end{bmatrix}, \tag{2.11}$$

can be calculated from measured signals and known parameters. The unknown parameters θ can be estimated using RLS which are presented next.

2.2 Recursive least squares

The least squares method is a common method when estimating unknown parameters. The method states that the parameter θ should be chosen to minimize the loss function

$$V(\theta, n) = \frac{1}{2}\sum_{i=1}^{n}(y(i) - \phi^T(i)\hat{\theta})^2, \tag{2.12}$$

where y is the observed variable, θ is the parameters to be estimated, and ϕ are known functions from the model. Solving for minimizing parameters the closed form solution is given as [3]

$$\hat{\theta} = \Big(\sum_{i=1}^{n}\phi(i)\phi^T(i)\Big)^{-1}\Big(\sum_{i=1}^{n}\phi(i)y(i)\Big). \tag{2.13}$$

When estimating parameters in real-time, it is desirable to make the calculations recursively. Updating the estimates when new data is available is more computationally efficient. The recursive form is given by

$$\hat{\theta}(k) = \hat{\theta}(k-1) + L(k)\Big(y(k) - \phi^T(k)\hat{\theta}(k-1)\Big), \tag{2.14}$$

where

$$L(k) = P(k)\phi(k) = P(k-1)\phi(k)\Big(I + \phi^T(k)P(k-1)\phi(k)\Big)^{-1} \tag{2.15}$$

and

$$P(k) = \Big(I - L(k)\phi^T(k)\Big)P(k-1). \tag{2.16}$$

$P(k)$ is referred to as the covariance matrix and $L(k)$ is referred to as the update gain.

As seen from (2.14) the standard form RLS can be seen as a filter that averages the data to give an estimate. This is a good strategy if the parameters are constant. In this case, at least the road grade is time-varying and therefore can not be tracked with the standard form RLS. In section 2.2.1 it is described how the RLS method can be modified to be able to track time-varying parameters.

2.2.1 Recursive least squares with forgetting factor

In the standard form recursive least squares method, the parameters are assumed to be constant. If the parameters are time-varying a forgetting factor λ can be introduced, which gives more weight to newer data. The loss function (2.12) combined with a forgetting factor is defined as [3]

$$V(\hat{\theta}, k) = \frac{1}{2}\sum_{i=1}^{k}\lambda^{k-i}\Big(y(i) - \phi^T(i)\hat{\theta}(k)\Big)^2, \tag{2.17}$$

where λ is the forgetting factor, which is $0 < \lambda \leq 1$ [18]. The same update equation (2.14) as in the recursive form is used, but with update gain $L(k)$ and covariance matrix $P(k)$ as

$$L(k) = P(k-1)\phi(k)\Big(\lambda + \phi^T(k)P(k-1)\phi(k)\Big)^{-1} \tag{2.18}$$

and

$$P(k) = \Big(I - L(k)\phi^T(k)\Big)P(k-1)\frac{1}{\lambda}. \tag{2.19}$$

2.2.2 Recursive least squares with multiple forgetting factors

In the standard recursive least squares method with a single forgetting factor, it is assumed that the parameters vary with similar rates. When two parameters vary with different rates one can introduce multiple forgetting factors to be able to track multiple parameters with greater accuracy [18]. In this case, the mass and the road grade vary with different rates. With multiple forgetting factors the covariance matrix is given as:

$$P(k) = \Lambda^{-1}\Big(I - L(k)\phi^T(k)\Big)P(k-1)\Lambda^{-1}, \tag{2.20}$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2)$, and λ_1, λ_2 are the forgetting factor for the first and second parameter. By using two different forgetting factors, the difference in varying rate can be incorporated in the recursive least squares method. This also gives more degrees of freedom in the update gain $L(k) = [L_1(k), L_2(k)]$.

For the special case, with estimation of two unknown parameters, we define the loss function as [18]

$$\begin{aligned} V(\hat{\theta}_1(k), \hat{\theta}_2(k), k) = \frac{1}{2}\sum_{i=1}^{k} \lambda_1^{k-i}\Big(y(i) - \phi_1(i)\hat{\theta}_1(k) - \phi_2(i)\theta_2(i)\Big)^2 + \\ \frac{1}{2}\sum_{i=1}^{k} \lambda_2^{k-i}\Big(y(i) - \phi_1(i)\theta_1(i) - \phi_2(i)\hat{\theta}_2(k)\Big)^2. \end{aligned} \tag{2.21}$$

The first term on the right hand side of the equation describes the error of the step k due to the first parameter estimate, $\hat{\theta}_1(k)$, and the second term describes the error of the step for the second parameter, $\hat{\theta}_2(k)$. In (2.21), λ_1, corresponds to the forgetting factor for the first parameter and λ_2 corresponds to the forgetting factor for the second parameter. The optimal estimates minimizes the loss function and are obtained as [18]

$$\frac{\partial V}{\partial \hat{\theta}_1(k)} = 0 \implies \sum_{i=1}^{k} \lambda_1^{k-1}(-\phi_1(i))\Big(y(i) - \phi_1(i)\hat{\theta}_1(k) - \phi_2(i)\theta_2(i)\Big) = 0. \tag{2.22}$$

By rearranging equation (2.22) we get

$$\hat{\theta}_1(k) = \Big(\sum_{i=1}^{k} \lambda_1^{k-i}\phi_1(i)^2\Big)^{-1}\Big(\sum_{i=1}^{k} \lambda_1^{k-i}(y(i) - \phi_2(i)\theta_2(i))\Big), \tag{2.23}$$

and when doing the same operations for $\hat{\theta}_2$ we obtain

$$\hat{\theta}_2(k) = \Big(\sum_{i=1}^{k} \lambda_2^{k-i}\phi_2(i)^2\Big)^{-1}\Big(\sum_{i=1}^{k} \lambda_2^{k-i}(y(i) - \phi_1(i)\theta_1(i))\Big). \tag{2.24}$$

By using the same methodology available between (2.23) and (2.24) and the classical form (2.13) the recursive form can be deduced:

$$\hat{\theta}_1(k) = \hat{\theta}_1 + L_1(k)\Big(y(k) - \phi_1(k)\hat{\theta}_1(k-1) - \phi_2(k)\theta_2(k)\Big), \tag{2.25}$$

where

$$L_1(k) = P_1(k-1)\phi_1(k)\Big(\lambda_1 + \phi_1^T(k)P_1(k-1)\phi_1(k)\Big)^{-1} \tag{2.26a}$$

$$P_1(k) = \Big(I - L_1(k)\phi_1^T(k)\Big)P_1(k-1)\frac{1}{\lambda_1}. \tag{2.26b}$$

Similarly for $\hat{\theta}_2$ we obtain

$$\hat{\theta}_2(k) = \hat{\theta}_2 + L_2(k)\Big(y(k) - \phi_1(k)\theta_1(k) - \phi_2(k)\hat{\theta}_2(k-1)\Big), \tag{2.27}$$

where

$$L_2(k) = P_2(k-1)\phi_2(k)\Big(\lambda_2 + \phi_2^T(k)P_2(k-1)\phi_2(k)\Big)^{-1} \tag{2.28a}$$

$$P_2(k) = \Big(I - L_2(k)\phi_2^T(k)\Big)P_2(k-1)\frac{1}{\lambda_2}. \tag{2.28b}$$

By replacing $\theta_1(k)$ and $\theta_2(k)$ with their estimates $\hat{\theta}_1(k)$ and $\hat{\theta}_2(k)$, which is justified when the estimate and the actual value is very close or within the region of convergence for the algorithm, we get [18]

$$\hat{\theta}_1(k) + L_1(k)\phi_2(k)\hat{\theta}_2(k) = \hat{\theta}_1(k-1) + L_1(k)\Big(y(k) - \phi_1(k)\hat{\theta}_1(k-1)\Big) \tag{2.29}$$

$$\hat{\theta}_2(k) + L_2(k)\phi_1(k)\hat{\theta}_1(k) = \hat{\theta}_2(k-1) + L_2(k)\Big(y(k) - \phi_2(k)\hat{\theta}_2(k-1)\Big), \tag{2.30}$$

which has the solution

$$\begin{bmatrix} \hat{\theta}_1(k) \\ \hat{\theta}_2(k) \end{bmatrix} = \begin{bmatrix} 1 & L_1(k)\phi_2(k) \\ L_2(k)\phi_1(k) & 1 \end{bmatrix}^{-1} \begin{bmatrix} \hat{\theta}_1(k-1) + L_1(k)\Big(y(k) - \phi_1(k)\hat{\theta}_1(k-1)\Big) \\ \hat{\theta}_2(k-1) + L_2(k)\Big(y(k) - \phi_2(k)\hat{\theta}_2(k-1)\Big) \end{bmatrix}. \tag{2.31}$$

The determinant of the term

$$\begin{bmatrix} 1 & L_1(k)\phi_2(k) \\ L_2(k)\phi_1(k) & 1 \end{bmatrix}$$

is always non-zero due to the fact that P_1 and P_2 are always positive. Therefore, the inverse of the matrix always exists.

By defining $L_{new}(k)$ as

$$L_{new}(k) = \frac{1}{1 + \frac{P_1(k-1)\phi_1(k)^2}{\lambda_1} + \frac{P_2(k-1)\phi_2(k)^2}{\lambda_2}} \begin{bmatrix} \frac{P_1(k-1)\phi_1(k)}{\lambda_1} \\ \frac{P_2(k-1)\phi_2(k)}{\lambda_2} \end{bmatrix} \tag{2.32}$$

we can rewrite (2.31) in the same form as (2.14) as

$$\hat{\theta}(k) = \hat{\theta}(k-1) + L_{new}(k)\Big(y(k) - \phi^T(k)\hat{\theta}(k-1)\Big). \tag{2.33}$$

By using this RLS with multiple forgetting factors both parameters can be tracked with greater accuracy, even though they vary with different rates.

The effective sample size, $w_{N,\lambda}$, approximates the number of observations that we are averaging over and for $\lambda \in (0, 1]$ then [4]

$$w_{N,\lambda} = \sum_{i=1}^{N} \lambda^{N-i} = \frac{1-\lambda^N}{1-\lambda}, \tag{2.34}$$

and when $N \to \infty$

$$w_{\infty,\lambda} = \frac{1}{1-\lambda}. \tag{2.35}$$

From the equations we can see that if $\lambda = 1$ then $w_{N,\lambda} = N$ and if $\lambda = 0$ then $w_{N,\lambda} = 1$. Generally λ is chosen as $0.9 \leq \lambda \leq 1$ [7]. If $\lambda = 0.9$ is chosen then $w_{N,\lambda} = 10$, and if $\lambda = 0.99$ is chosen then $w_{N,\lambda} = 100$.

2.3 Evaluation metrics

During the parameter estimation and mass estimation, the algorithm will be evaluated using different metrics. One of the measures used is root mean square error, RMSE, which is calculated as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{2.36}$$

where $\hat{y}$ is the predicted signal and y is the measured signal. Another measure that will be used is mean error percentage which is calculated as

$$MEP = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{2.37}$$

In previous literature, the mass estimate is achieved with an accuracy of 5-10 %, and therefore the time within 5 % of the true mass will also be a measure that will be taken into consideration. For all tests, these metrics will be started when the velocity is larger than 0 and then calculated for the whole duration of the test. This is done due to the different amount of time between the start of the data collection and when the vehicle starts moving for the different tests should not affect the result.

3

Estimator algorithm

This chapter describes the implementation of the estimator algorithm. This includes how the vehicle longitudinal model is combined with the RLS method. How the resampling and signal filtering is also presented. An explanation of the longitudinal motion detector is also presented in this chapter.

3.1 Vehicle model combined with RLS

By combining the vehicle longitudinal model and the RLS method the parameters θ, the mass and road grade, can be estimated by using (2.10)-(2.11). An overview of the algorithm is shown in Figure 3.1.

The inputs F_{wheel}, v^2 and $\dot{v}$ are available on the CAN bus and ρ, S, C_d, g, θ_μ are constants that can be determined, this is described in Chapter 4. The wheel torque τ_{wheel} is calculated in the electrical control unit in the engine and broadcast on the CAN bus. The wheel force is calculated as

$$F_{\text{wheel}} = \frac{\tau_{\text{wheel}}}{r_{\text{wheel}}}, \tag{3.1}$$

where r_{wheel} is the wheel radius. Therefore no estimation of the dynamics in the driveline is necessary, which is needed in previous literature. The RLS method estimates the parameter θ with the inputs F_{wheel}, v^2 and $\dot{v}$.

The initial guess to the RLS algorithm in all tests were

$$m_{\text{initial}} = m_{\text{curb}} + m_{\text{driver}} + m_{\text{fuel}}, \tag{3.2}$$

where $m_{\text{curb}} = 1421$ kg, $m_{\text{driver}} = 70$ kg, and m_{fuel} was calculated as the measured amount of fuel multiplied by 0.78 kg/l, which is the density of petrol.

The true mass of the vehicle was calculated as

$$m_{\text{true}} = m_{\text{curb}} + m_{\text{driver}} + m_{\text{fuel}} + m_{\text{load}}, \tag{3.3}$$

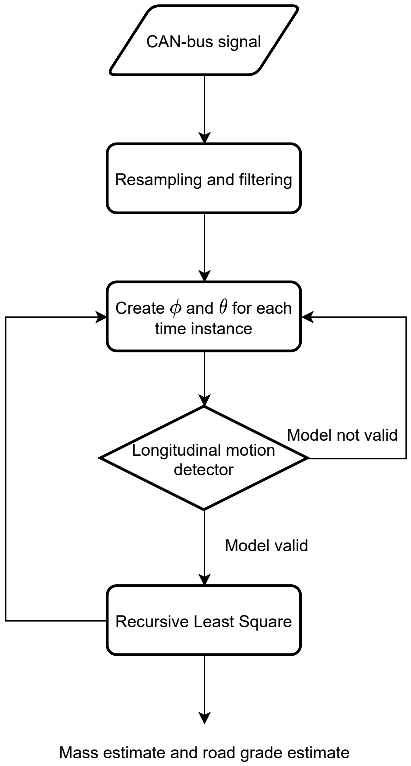

CAN-bus signal
Resampling and filtering
Create ϕ and θ for each time instance
Longitudinal motion detector
Model not valid
Model valid
Recursive Least Square
Mass estimate and road grade estimate

Figure 3.1: *Flow of the estimator algorithm.*

where m_{load} is the additional load in the vehicle.

Two different variants of the algorithm were analyzed, one with a single forgetting factor, SFF-RLS, and one with multiple forgetting factors, MFF-RLS.

3.2 Resampling and signal filtering

The signals are received from the CAN bus with different frequencies depending on the signal received, therefore a resampling is necessary. The resampling is done with a fixed frequency of 50 Hz. The last value of the signal between $t-1$ and t is saved at time t. Most of the signals arrive at approximately 50 Hz or lower, therefore not that much data is dropped.

After the resampling the signals are filtered. In the filtering a moving average filter is used to low-pass filter the signals. The span used is 10. The signals that are integers, as current gear and fuel level, are not filtered. An example of the moving average filter with span 3 is given by

$$\begin{aligned}
y(1) &= y(1) \\
y(2) &= (y(1) + y(2))/2 \\
y(3) &= (y(1) + y(2) + y(3))/3 \\
y(4) &= (y(2) + y(3) + y(4))/3 \\
&\vdots
\end{aligned} \tag{3.4}$$

By using a moving average with a large window the fast dynamics in the signal will not be detectable, but the signal will be much smoother. When using a large window it will filter out fast dynamics in the signals that can occur due to irregularities in the road as potholes and similar. It is also done to filter out noise in the signals.

3.3 Longitudinal motion detector

When the signals have been resampled and filtered the mass and road grade are estimated simultaneously with the recursive least squares method as described in section 2.2. The estimation is only done when several conditions are met. When these conditions are not met the vehicle longitudinal model is not valid due to other forces acting on the vehicle. For example, when having a large lateral acceleration other forces are acting on the vehicle than described in section 2.1. When the longitudinal motion detector is turned off no estimation is done but the result is maintained until the next time the motion detector is turned on.

The estimation is done when all these conditions are met:

- No gear shift is in progress.
- The absolute value of the lateral acceleration is smaller than 0.5 m/s^2
- The absolute value of the longitudinal acceleration is larger than 0.3 m/s^2.
- The velocity is larger than 15 km/h.
- No brake is applied.

The first condition is inspired by [18] where the authors show that the algorithm performs better when turned off during gear shifts. During gear shifts, the transmission disengages which means that the power transmitted to the wheels is reduced and for a short moment no power is transmitted to the wheels. This makes the wheel torque change quickly during gear shifts which affects the estimator. The time for a gear shift is short and therefore it is not necessary to run the estimator during gear shifts.

The second and third conditions are because during low speeds or high lateral movement the longitudinal model is not valid due to other forces being applied that affect the dynamics of the vehicle. These conditions were found after analysis of the physical model and the values in these conditions were found after investigating the performance of the estimator algorithm. The values were also chosen to ensure that the motion is predominantly longitudinal.

The fourth condition was inspired by [6], where it is discussed that the vehicle model works well when the speed is not very close to 0 km/h.

The last condition is because the brake force is difficult to estimate and therefore no estimation is done during braking.

4

Results and discussion

This chapter presents what experiments were performed and how the data was collected. It also presents the results of the estimation of the constant parameters. The results and performance of the estimation algorithm developed are shown in this chapter. A sensitivity analysis is also performed and presented.

4.1 Experiments and data collection

Data were collected at two different times, the first time during snow conditions and the second during dry road conditions. Test numbers 1-5 were done during the first test run and the remaining tests during the second test run. Test 1 was done 10 times and test 2 was done 5 times, all other tests were done by driving the vehicle for 30 minutes in traffic. A description of all tests is presented in table 4.1. During the tests, all data sent on the CAN bus were collected and saved into a text file.

All tests are constructed to reflect typical scenarios for driving a passenger vehicle. All tests are performed on an open road with traffic, which makes the experiments realistic. The tests are split into three different typical driving scenarios, country roads, city roads, and highways. This is done to investigate if the algorithm performs better in any scenario or if it is not possible to use during certain scenarios. In these scenarios, several different tests are performed. In all these scenarios there are similar tests done but with different extra loads in the vehicle, in order to see how the algorithm reacts to different weights of the vehicle. This also reflects typical real-life scenarios, where 200 kg extra load in the vehicle roughly corresponds to driving with 3 passengers in the vehicle.

For country roads, the tests consist of three tests with regular driving, one without additional weight, one with 200 kg extra load, and one with 400 kg extra load. There is also one test where the speed is constantly varied between 70 –

80 km/h, to have lots of excitation of the inputs.

The city road scenarios are split into three different tests, all of which consist of regular driving in the city. The first test is without any additional weight, the second with 200 kg extra load and the last with 400 kg extra load.

The highway scenario is split into two tests, one with regular driving on a highway without any additional load and one with 400 kg extra load.

Table 4.1: *Description of all tests where data was collected.*

Test number	Type of road	Description	Extra load (kg)
Test run 1			
1	Coast-Down	Used when determining drag coefficient and rolling resistance. During this test 10 coast-down scenarios was done.	0
2	Country roads	Acceleration from 0-80 km/h, done 5 times.	0
3	Country roads	Driving during normal traffic conditions on country roads with speed varied from 70-80 km/h.	0
4	Country roads	Driving during normal traffic conditions on country roads.	0
5	City roads	Driving during normal traffic conditions on city roads.	0
Test run 2			
6	Country roads	Driving during normal traffic conditions on country roads.	200
7	Country roads	Driving during normal traffic conditions on country roads.	400
8	City roads	Driving during normal traffic conditions on city roads.	200
9	City roads	Driving during normal traffic conditions on city roads.	400
10	Highway	Driving during normal traffic conditions on highway.	0
11	Highway	Driving during normal traffic conditions on highway.	400

4.2 Parameter identification

Before estimating the unknown parameters, mass and road grade, the constant parameters in the model, described in (2.11), need to be determined. The constant parameters are the rolling resistance C_r, the surface area S, and the drag coefficient C_d. These parameters are determined by doing coast-down scenarios. One coast-down scenario consists of accelerating to a high speed, putting the car in neutral gear, and letting the car coast down to low speed. This is done on flat ground, where the road grade is close to 0 degrees. The data was logged and the different segments that contain the 10 coast-down scenarios were merged into a file for later analysis. Using (2.11), y and ϕ can be created as

$$\phi = \begin{bmatrix} -mg & \frac{1}{2}\rho v_1^2 \\ -mg & \frac{1}{2}\rho v_2^2 \\ \vdots & \vdots \\ -mg & \frac{1}{2}\rho v_n^2 \end{bmatrix}, y_{coast} = \begin{bmatrix} a_1 m \\ a_2 m \\ \vdots \\ a_n m \end{bmatrix}, \tag{4.1}$$

from known constants and measured signals. From this we can get the estimates as

$$C_{est} = R_N^{-1} F_N \tag{4.2}$$

$$R_N = \phi^T \phi \tag{4.3}$$

$$F_N = \phi^T y_{coast}, \tag{4.4}$$

where $C_{est} = \begin{pmatrix} \hat{C}_r \\ \hat{C}_{sd} \end{pmatrix}$. By combining the drag coefficient and surface area we can estimate them as one variable as $\hat{C}_{sd} = S\hat{C}_d$. The covariance matrix of C_{est} is calculated as

$$P_N = \text{Cov}(\hat{\theta}_N) = \frac{1}{N}\hat{\sigma}_N R_N, \tag{4.5}$$

where

$$\hat{\sigma}_N = V_N(\hat{C}_{est}) = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \phi^T(t)\hat{\theta}_N\right)^2, \tag{4.6}$$

which will be used to determine the insecurity of the estimated parameters C_{est}. The diagonal elements of P_N is the variance of each parameter.

The vehicle was weighed to get the curb weight, where the amount of fuel was measured before weighing the car. The curb weight is constant for each car model and only needs to be measured one time. The air density, ρ, was calculated by measuring the ambient temperature and getting the air pressure and relative humidity from SMHI. This value was calculated during the first test run. During the year typical values for the air density varies between 1.15-1.35 [1]. The wheel radius was measured by hand and can be seen as constant as long as the wheel is not changed. The used constants is shown in Table 4.2.

The constant parameters C_{est} were estimated by using (4.2)-(4.4). The segments of the test where coast-down were performed were merged into one file that only contains these coast-down scenarios. The different segments is marked in Figure 4.1. The estimated parameters are

$$C_{est} = \begin{pmatrix} \hat{C}_r \\ \hat{C}_{sd} \end{pmatrix} = \begin{pmatrix} 0.0103 \\ 1.0512 \end{pmatrix}. \tag{4.7}$$

Table 4.2: *Values of the constant parameters.*

Description	Constant	Value	Uncertainty
Curb weight of the car	m_{curb}	1421 kg	± 0.5 kg
Air density	ρ	1.31 kg/m^3	± 0.02 kg/m^3
Wheel radius	r_{wheel}	0.358 m	± 0.01 m
Gravitational acceleration	g	9.81 m/s^2	

The covariance matrix of the parameters is

$$P_N = 10^{-4}\begin{pmatrix} 0.0001 & -0.0016 \\ -0.0016 & 0.1284 \end{pmatrix}, \tag{4.8}$$

which indicates that the certainty of the estimation of C_{est} is high.

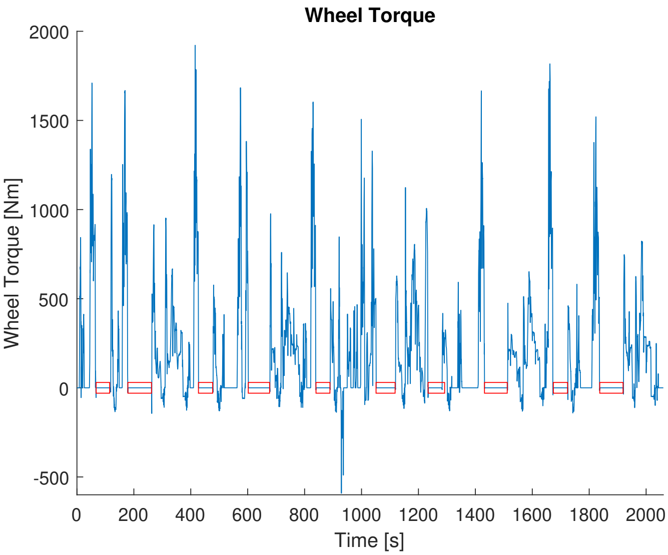


Figure 4.1: *Measurement of wheel torque during coast-down scenarios with the coast-down marked in red.*

The coast-down scenario was performed during the first test run. There was not a possibility to do coast-down scenarios during the second test run due to the covid-19 pandemic. Due to the different road conditions between the first test run and the second test run, another value for C_r was chosen for the second set of tests. The value of C_{sd} is assumed to be the same for all tests, and for the second test run the rolling resistance was chosen as $C_r = 0.02$ which is a typical value for passenger cars on dry asphalt [13].

4.3 Mass estimation

In this section, the results for the mass estimation for the different tests are presented. The mass was estimated using the SFF-RLS and the MFF-RLS. Plots for mass estimate and road grade estimate for all tests can be found in Appendix B.

4.3.1 Single forgetting factor

With a single forgetting factor for both the mass estimate and the grade, it is difficult to track both parameters as they vary with different rates, as described in Section 2.2. The average MEP for all tests for different λ is shown in Figure 4.2, where the MEP is calculated for each test and then the average MEP is calculated. From the figure it can be seen that $\lambda = 1$ gives the best result.

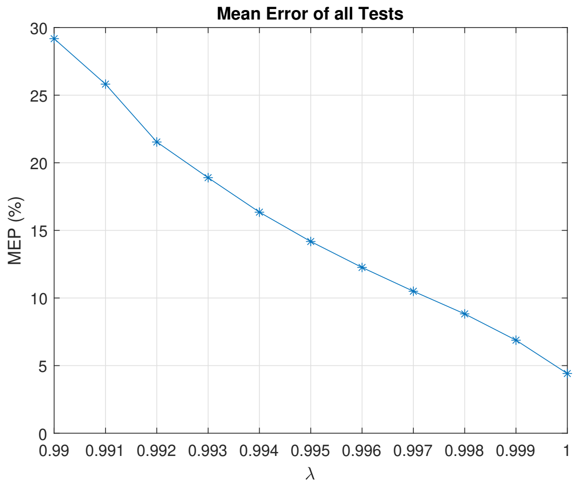


Figure 4.2: *Average MEP for all tests with different λ.*

When using RLS with a forgetting factor of 1 the road grade is assumed to be constant which makes the estimate near-constant. The mass estimate and road grade estimate for test 4 can be seen in Figure 4.3. From the figure, it can be seen that the grade estimate does not vary as much as can be expected but the algorithm gives a good estimate of the mass. The mass estimate is within 5 % of the true mass after 60 seconds after the vehicle starts moving and stays within the limit the whole test. The estimate changes during the duration of the test, but converges towards the true mass. In Table 4.3 the results for all tests can be seen. The algorithm estimates the mass within 10 % of the true mass in all tests, and within 5 % of the true mass in 5 of the tests. The largest mean error is in test 2 where the mean error is 8.08 %. When looking at the average error over all tests it is 4.42 %.

When lowering the forgetting factor but keeping it near 1, the grade estimate can improve but the variance of the mass estimate is large. Test 4 with a forgetting

Table 4.3: *Results for the different tests with a single forgetting factor.*

Test	RMSE	MEP (%)	Percentage of time within 5 % of the true mass (%)
2	128.83	8.08	12.57
3	27.46	1.41	99.14
4	38.01	1.58	97.29
5	103.13	6.61	1.17
6	92.76	5.33	27.71
7	138.55	6.82	22.64
8	64.77	3.5	94.52
9	106.01	5.19	45.32
10	28.93	1.22	96.82
11	95.71	4.43	64.01

factor of $\lambda = 0.995$ can be seen in Figure 4.4. As described in Section 2.2.1 a forgetting factor of $\lambda = 0.995$ corresponds to that approximately 200 samples is used in the estimate. With the sampling frequency of 50 Hz, this corresponds to 4 seconds of data.

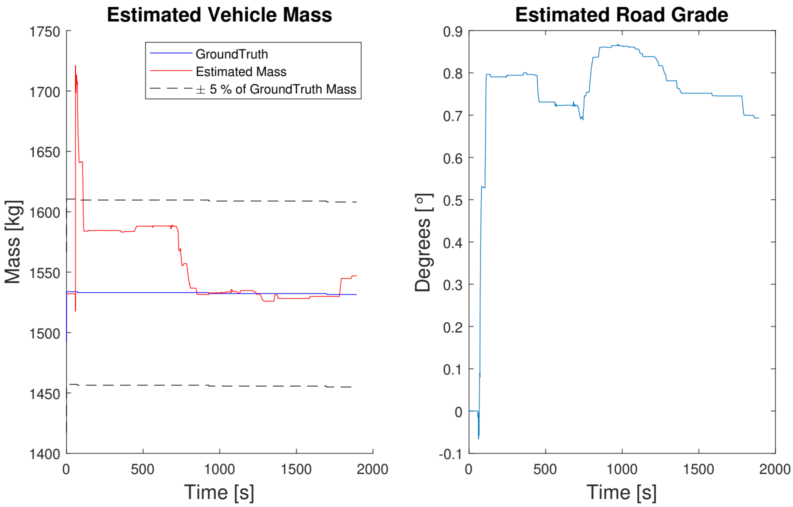


Figure 4.3: *Mass and road grade estimate during test 4 with a single forgetting factor,* $\lambda = 1$.

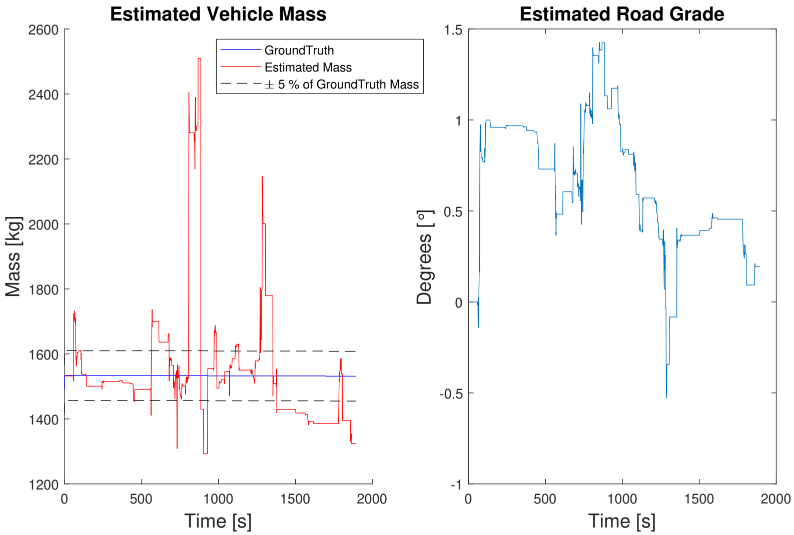


Figure 4.4: *Mass and road grade estimate during test 4 with a single forgetting factor,* $\lambda = 0.995$

Discussion

As can be seen in Table 4.3 the RLS algorithm with a single forgetting factor gives a good estimate for the mass in all tests and is within 10 % of the true mass in all tests. The road grade estimate is small and does not vary as much when using $\lambda = 1$ but the mass estimate converges towards the true mass. As the road grade estimate is close to 0 degrees and is estimated with $\lambda = 1$ another suitable approach would be to not estimate the road grade but to use a value of 0 for the road grade during the whole test. When estimating the road grade with λ the road grade will be near-constant which does not reflect the real-world conditions. As seen in the average MEP for all tests the estimator gives a good mass estimate when using $\lambda = 1$ for both the mass and the road grade, and therefore we can estimate the road grade and mass simultaneously.

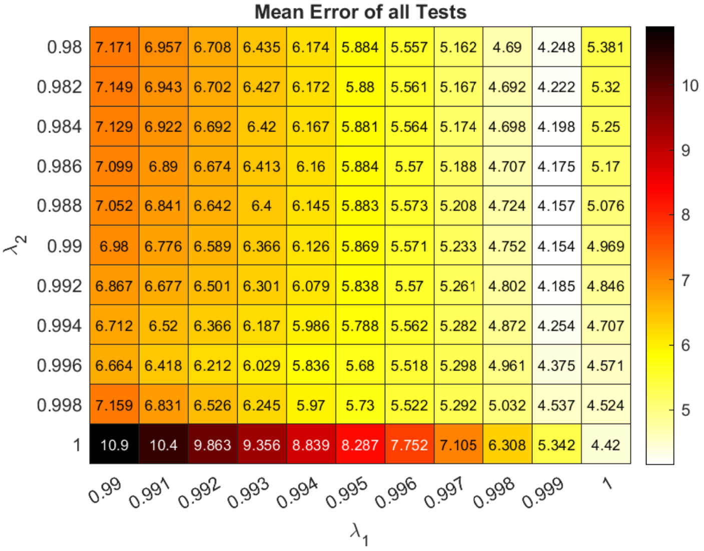
If driving on a road with a large road grade, or a road grade that varies much, the algorithm would estimate the force induced by the road grade, seen in (2.5), too small compared to the actual force affecting the vehicle. The mass estimate could thus become far from the true mass as the road grade estimate would still be very small. Another aspect that is worth noting is that when using $\lambda = 1$ for the road grade the algorithm will be giving different estimations if the vehicle is driven in a slope during the first minute of estimation or after a long time, given that the algorithm is taking all measurements into consideration. This could make the estimate far from the true mass if there is a large slope at the beginning of the estimation, and as $\lambda = 1$, this estimate would affect the estimation for the whole duration of the run.

It takes some time before the algorithm gives a good estimate, around one minute. The error is relatively small for all tests, and therefore this approach could be suitable when implementing in a real vehicle.

When using $\lambda < 1$ the mass estimate varies a lot and is therefore not a suitable approach when estimating the mass. When using $\lambda = 0.995$ approximately 4 seconds of data is used, and as the mass changes very slowly this value is too low when estimating the mass. The mass only changes a few kilograms during the whole test run but the road grade change more frequently and for this parameter a forgetting factor of $\lambda = 0.995$ is more realistic, and could be a more suitable approach if the road grade is large or varies a lot. As seen in Figure 4.4 the mass estimate, in this case, becomes far from the true value. This, combined with the results shown in Figure 4.2, indicates that the value of λ is more important to choose so that it matches the varying rate of the mass rather than the road grade. Therefore it is reasonable to choose a forgetting factor of $\lambda = 1$ in our case, where the road grade is not that varying.

4.3.2 Multiple forgetting factors

As mentioned in section 2.2.1 a way to more accurately track both the mass and the road grade is to introduce multiple forgetting factors, one for each parameter, where λ_1 is associated with the mass and λ_2 is associated with the road grade. First, the MEP was calculated for each combination of λ_1 and λ_2, where λ_1 was kept near 1 as the mass is a near-constant parameter. As mentioned in Section 2.2.1 the forgetting factor is generally chosen within $0.9 \leq \lambda \leq 1$, and therefore λ_2 was varied between 0.9 and 1. When examining the results $\lambda_2 < 0.98$ did not give good results and therefore $0.98 \leq \lambda_2 \leq 1$ was examined further. The average MEP for all tests is shown in Figure 4.5. Note that for $\lambda_1 = 1$, $\lambda_2 = 1$ we get classical RLS.



λ_2 \ λ_1	0.99	0.991	0.992	0.993	0.994	0.995	0.996	0.997	0.998	0.999	1
0.98	7.171	6.957	6.708	6.435	6.174	5.884	5.557	5.162	4.69	4.248	5.381
0.982	7.149	6.943	6.702	6.427	6.172	5.88	5.561	5.167	4.692	4.222	5.32
0.984	7.129	6.922	6.692	6.42	6.167	5.881	5.564	5.174	4.698	4.198	5.25
0.986	7.099	6.89	6.674	6.413	6.16	5.884	5.57	5.188	4.707	4.175	5.17
0.988	7.052	6.841	6.642	6.4	6.145	5.883	5.573	5.208	4.724	4.157	5.076
0.99	6.98	6.776	6.589	6.366	6.126	5.869	5.571	5.233	4.752	4.154	4.969
0.992	6.867	6.677	6.501	6.301	6.079	5.838	5.57	5.261	4.802	4.185	4.846
0.994	6.712	6.52	6.366	6.187	5.986	5.788	5.562	5.282	4.872	4.254	4.707
0.996	6.664	6.418	6.212	6.029	5.836	5.68	5.518	5.298	4.961	4.375	4.571
0.998	7.159	6.831	6.526	6.245	5.97	5.73	5.522	5.292	5.032	4.537	4.524
1	10.9	10.4	9.863	9.356	8.839	8.287	7.752	7.105	6.308	5.342	4.42

Figure 4.5: *Average mean error for different combinations of λ_1 and λ_2.*

From Figure 4.5 it can be seen that the combination that gives the lowest average error is $\lambda_1 = 0.999$ and $\lambda_2 = 0.99$. In Figure 4.6 the mass and road grade estimate is shown for test 6, with the optimal forgetting factors. From the figure, it can be seen that the algorithm gives a reasonable mass estimate within 25 seconds. The mass estimate changes during the whole duration of the test but is within 5 % of the true mass for the whole duration.

As described in Section 2.2.1, when using a forgetting factor of 1 the parameter is assumed to be constant. As the mass estimate is near-constant an approach of using $\lambda_1 = 1$ is also considered. As shown in Figure 4.5 a value of $\lambda_1 = 1$, $\lambda_2 = 0.99$ gives a slightly larger error on average than the optimal forgetting factors. In Figure 4.7 the mass and road grade estimation for test 6 is shown, with a forgetting factor of 1 for the mass and 0.99 for the road grade. The estimate is around 1 % of the true mass during the whole test, and changes very little during the test, compared to Figure 4.6. The algorithm gives a good estimate after 15 seconds, and the estimate changes very little after the initial 15 seconds of the test.

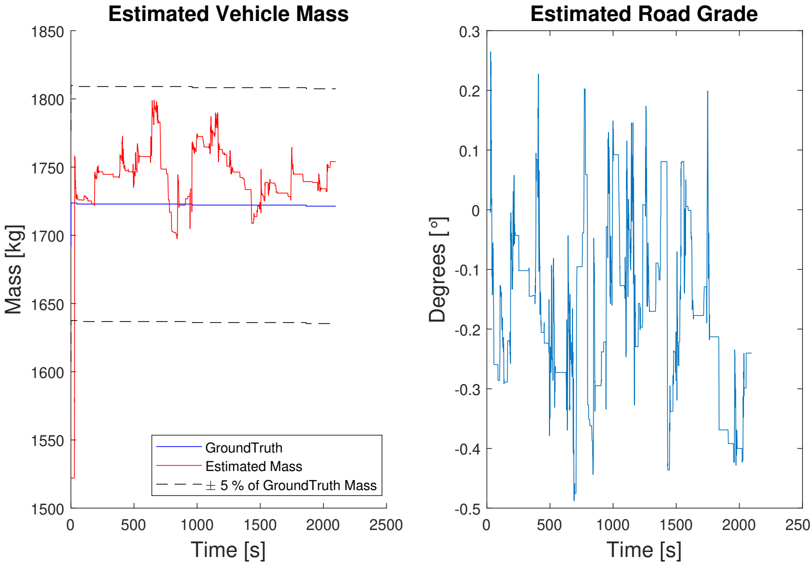


Figure 4.6: *Mass and road grade estimate during test 6 with* $\lambda_1 = 0.999$ *and* $\lambda_2 = 0.99$.

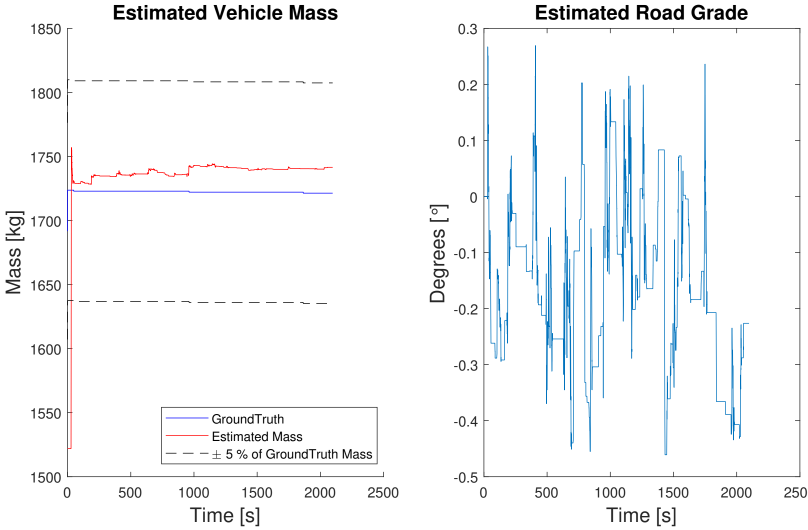


Figure 4.7: *Mass and road grade estimate during test 6 with* $\lambda_1 = 1$ *and* $\lambda_2 = 0.99$.

The results for all tests are listed in Table 4.4 and Table 4.5 for the two different sets of forgetting factors. The algorithm with $\lambda_1 = 1$, $\lambda_2 = 0.99$ estimates the mass quickly but when overestimating or underestimating it does not change towards the true mass as the algorithm with a single forgetting factor does. An example of this can be seen in Figure 4.8, where the estimate is too high and converges towards the true mass very slowly. For all tests, both sets of forgetting factors estimate the mass very quickly. The average error for all tests are 4.15 % when using $\lambda_1 = 0.999$ and 4.97 % when using $\lambda_1 = 1$.

Table 4.4: *Results for the different tests with forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

Test	RMSE	MEP (%)	Percentage of time within 5 % of the true mass (%)
2	133.85	8.58	1.06
3	57.16	3.42	82.70
4	98.54	4.89	61.35
5	128.19	8.05	1.07
6	33.64	1.43	99.12
7	128.99	5.72	37.42
8	35.01	1.51	99.05
9	68.59	2.56	90.38
10	22.01	0.92	96.82
11	93.42	4.45	74.76

Table 4.5: *Results for the different tests with forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

Test	RMSE	MEP (%)	Percentage of time within 5 % of the true mass (%)
2	124.20	8.03	1.06
3	71.09	4.59	58.69
4	144.08	9.28	1.52
5	156.37	10.09	0.87
6	24.94	1.02	99.12
7	101.90	4.88	56.73
8	37.19	1.82	99.05
9	83.14	3.40	77.64
10	36.30	2.25	97.97
11	82.54	3.94	98.85

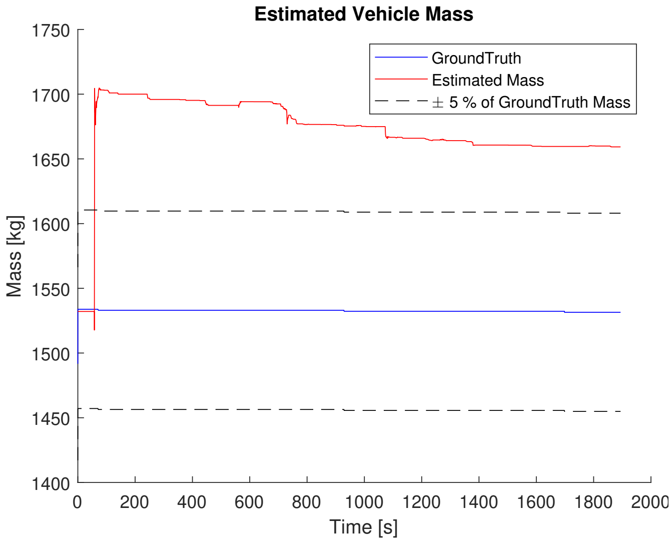


Figure 4.8: *Mass estimate during test 4 with $\lambda_1 = 1$ and $\lambda_2 = 0.99$.*

Discussion

When using MFF-RLS with $\lambda_1 = 1$ the mass estimate converges quickly, but in the case of estimating the mass wrong, the estimate does not change towards the true mass. Both sets of MFF-RLS gives a good estimate in most of the tests, and the error is within 10 % in all tests but test 5 for $\lambda_1 = 1$, $\lambda_2 = 0.99$ where the error is 10.09 %. As can be seen, when examining the time within 5 % of the true mass the MFF-RLS with $\lambda_1 = 1$ either gives an estimate quickly within the margin or the estimate stays outside of the margin for the whole duration of the run. This is because when using $\lambda_1 = 1$ all samples are taken into consideration in the estimate. When comparing this to the SFF-RLS it can be seen that the SFF-RLS changes more during the run, and this depends on the road grade estimate that is very small in the SFF-RLS but varies more in the MFF-RLS.

When using $\lambda_1 = 0.999$ the estimate changes more during the whole test, this is because the algorithm approximately uses the last 1000 samples, which corresponds to around 20 seconds of driving. This makes the algorithm able to converge towards the true mass, even if the initial estimate is far from the true mass. When comparing this to the SFF-RLS one can see that the estimate from the MFF-RLS with $\lambda_1 = 0.999$ changes more than the SFF-RLS during the run, which is because the forgetting factor for the mass is lower than 1. The MFF-RLS with $\lambda_1 = 0.999$ gives a estimate within 15 seconds of driving.

Test 5 gives a larger error for both sets of λ_1, where the MEP is 8.05 % when using $\lambda_1 = 0.998$ and 10.09 % when using $\lambda_1 = 1$. When looking at the plots for this test it can be seen that after 50 seconds in the test both algorithms estimate the mass about 250 kg below the true mass. There is a big jump in the estimate around 48 seconds in the plot. The MFF-RLS with $\lambda_1 = 0.999$ gives an estimate that is moving towards the true value but the MFF-RLS with $\lambda_1 = 1$ stays approximately the same. The reason the algorithms give such a large error can depend on several factors such as wheel slip or a change of road conditions at the time 48 seconds where the big jump in the estimate is. When discarding the first 50 seconds of the test and starting the algorithm at $t = 50$ s the MEP is around 4 % for both sets of λ_1. This indicates that the type of driving scenario does not affect the estimation. It also indicates that there was some source of error either in the signals measured or there was a change in road conditions at that point that is the reason for the large MEP for the test.

4.3.3 Comparison of driving scenarios

In Table 4.6 the mean error for the three different driving scenarios, country roads, city roads, and highways, are presented. The mean error is calculated for SFF-RLS with $\lambda = 1$, MFF-RLS with $\lambda_1 = 0.999$, $\lambda_2 = 0.9$, and MFF-RLS with $\lambda_1 = 1$, $\lambda_2 = 0.99$. In country driving tests 3, 4, 6, and 7 are used, city driving consists of tests 5, 8, 9, and highway driving consists of tests 10 and 11.

Table 4.6: *Average error for different driving scenarios.*

Scenario	Average MEP (%)		
	SFF ($\lambda = 1$)	MFF ($\lambda_1 = 0.999$)	MFF ($\lambda_1 = 1$)
Country roads	3.79	3.87	4.94
City roads	5.10	4.04	5.10
Highway roads	2.83	2.69	3.10

From the table, it can be seen that all three algorithms estimate highway driving best. All three versions of the algorithm estimate the city roads the worst of the scenarios.

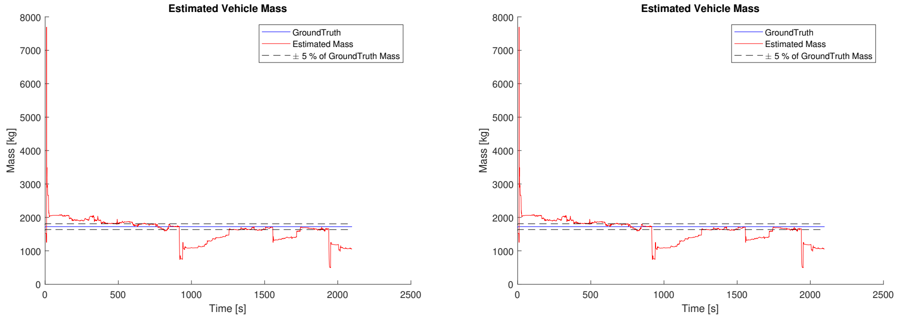
Discussion

All three algorithms estimate highway roads best but the average MEP is close between all different scenarios with an average MEP between 2.69 – 5.10 %. This indicates that the algorithms are robust to changes in driving scenarios, but in order to draw definitive conclusions more tests in each driving scenario would be needed. This also proves that the vehicle longitudinal model is fairly accurate. One aspect that needs to be taken into consideration is the different amount of tests in each scenario. There are only two tests on highways, which means that the average MEP for highway roads is quite uncertain and would need more tests in order to certainly conclude that the algorithms perform better in that scenario.

In [5] the authors discuss the general problem that the convergence of mass estimators is dependent on the persistence of excitation. This depends on driver aggressiveness and the type of driving scenario. When driving on city roads it is typically more excitation in the acceleration, but a smaller wheel force. When driving on highways there is typically smaller excitation in acceleration due to that the speed does not vary as much, but a larger wheel force is applied. The longitudinal motion detector makes sure that there is excitation in the inputs as the estimation is stopped when the acceleration or velocity is too low. The results indicate that the estimator works well for all driving scenarios and that the persistence of excitation is not a problem when using the longitudinal motion detector.

4.3.4 Estimation without longitudinal motion detector

As described in Section 3.3 the longitudinal vehicle model is not valid during certain maneuvers and during gear shifts. In Figure 4.9 the mass estimate during test 6 is shown, without the longitudinal motion detector, using MFF-RLS with $\lambda_1 = 0.999$, $\lambda_2 = 0.99$. The same estimate is shown in Figure 4.9, but with the motion detector activated. The MEP is 14.50 % without the motion detector and 1.43 % with the motion detector. Without the motion detector, the algorithm estimates the mass too high at the start of the test and the estimate slowly converges towards the true value, and at the end of the test, it estimates the mass too low. After 15 seconds the algorithm gives an estimate that are 2050 kg, and the true mass is 1723 kg.



*(**a**) Mass estimate without the longitudinal motion detector*

*(**b**) Mass estimate with the longitudinal motion detector*

Figure 4.9: *Mass estimate during test 6 with forgetting factor* $\lambda_1 = 0.999$, $\lambda_2 = 0.99$

Discussion

As described in Section 3.3 the model is not valid during gear shifts or lateral movement, and Figure 4.9 proves this point. As the error is around 80 % after the initial 15 seconds it can be concluded that the algorithm is in need of the longitudinal motion detector. As previously discussed the longitudinal motion detector also makes sure that there is excitation in the inputs which makes the estimator able to work.

4.4 Sensitivity analysis

In this section, a sensitivity analysis is presented. The analysis was done by varying one parameter at a time and calculating the mean error for each value of the parameter. The parameters that were varied were C_r, C_{sd} and r_{wheel}. This is done for the algorithm both with the single forgetting factor and with multiple forgetting factors for all tests. When using SFF-RLS the forgetting factor used throughout this section is $\lambda = 1$, when using MFF-RLS two sets of forgetting factors were used. The two sets are $\lambda_1 = 0.999$, $\lambda_2 = 0.99$, and $\lambda_1 = 1$, $\lambda_2 = 0.99$.

4.4.1 Rolling resistance

First, the sensitivity of the rolling resistance coefficient, C_r, was investigated. The coefficient was varied with all other parameters kept constant. A typical value of C_r varies between $0 - 0.03$ [15], and therefore C_r was varied between $0 - 0.05$ in order to determine the effects of a slightly higher value. The mean error was calculated for each C_r. This was done first for the algorithm with a single forgetting factor and then for the algorithm with multiple forgetting factors.

In Figure 4.10, the mean error for each test during the first test run is shown using SFF-RLS and the second test run is shown in Figure 4.11. Note that the used value is $C_r = 0.01$ for the first run and $C_r = 0.02$ for the second run. As can be seen in the figure, the rolling resistance has very little effect on the estimate. Between the lowest value of C_r and the highest the error of the estimate changes about 2 percentage points. For all different values of C_r, the mean error is within 10 % for all tests. When looking at the tests during the first test run it can be seen that for tests 3 and 4 the mean error is between 1-2 %, as can be seen in Figure 4.10. For test 2 and 5 different values of C_r also gives a small difference in mean error.

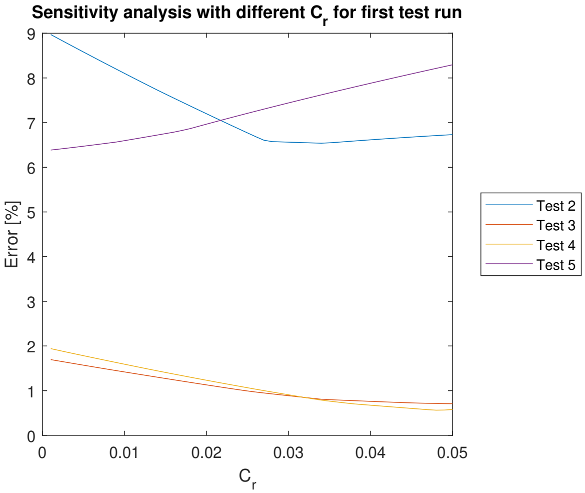


Figure 4.10: *Sensitivity analysis with different C_r for the tests during the first test run using SFF-RLS with $\lambda = 1$.*

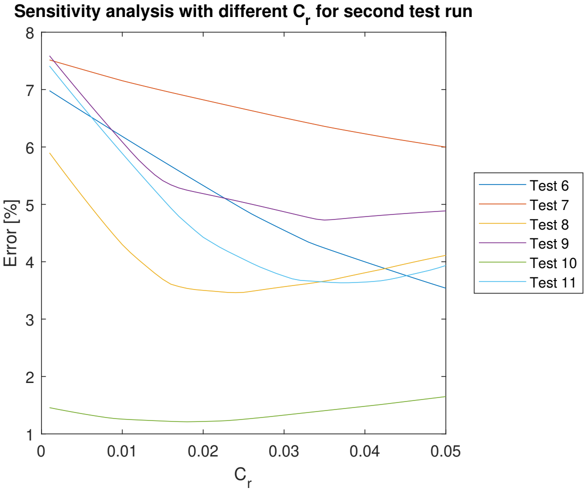


Figure 4.11: *Sensitivity analysis with different C_r for the tests during the first test run using SFF-RLS with $\lambda = 1$.*

In Figure 4.12 the mean error for the first test run, using $\lambda_1 = 0.999$, for the MFF-RLS is shown. The mean error when using $\lambda_1 = 1$ is shown in Figure 4.13. The mean error for the second test run is shown in Figure 4.14 and Figure 4.15 with the different sets of λ_1.

From the figures, it can be seen that during the first test run the coefficient can be changed up to $C_r = 0.05$ and the mean error is within 10 % for all tests but test 5 for both sets of λ_1. When using $\lambda_1 = 1$ the error changes more than for $\lambda_1 = 0.999$ when C_r changes. For the algorithm with $\lambda_1 = 1$ the error can become close to 0 whereas for $\lambda = 0.999$ the minimum error is about 1 % for test 3 and 4.

During the second test run, the same characteristics can be seen. When using $\lambda_1 = 0.999$ most tests have a minimum error for $C_r = 0.015 - 0.02$ and are within 10 % for a range $0 \leq C_r \leq 0.04$. Test 11 has a minimum value for $C_r = 0.027$ which is higher than the other tests. When comparing this to when using $\lambda_1 = 1$ it can be seen that the variation is larger for which C_r gives the minimum error. This can be seen in Figure 4.15. As for the first test run the minimum error can become very close to 0 %. For all tests C_r can change ±0.005 from the estimated value ($C_r = 0.02$), and stay within 10 % error.

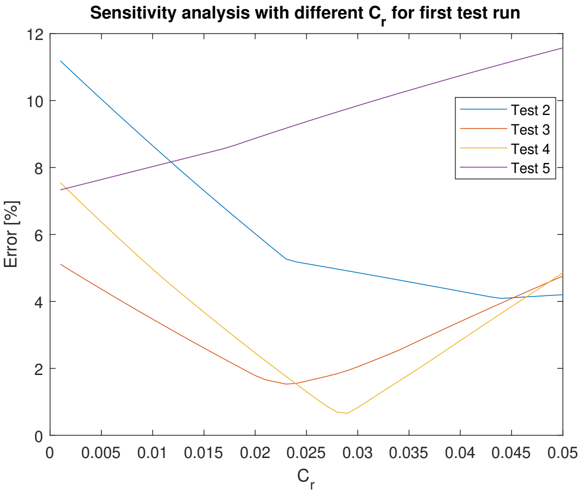


Figure 4.12: *Sensitivity analysis with different C_r using MFF-RLS for the tests during the first test run using $\lambda_1 = 0.999$ and $\lambda_2 = 0.99$.*

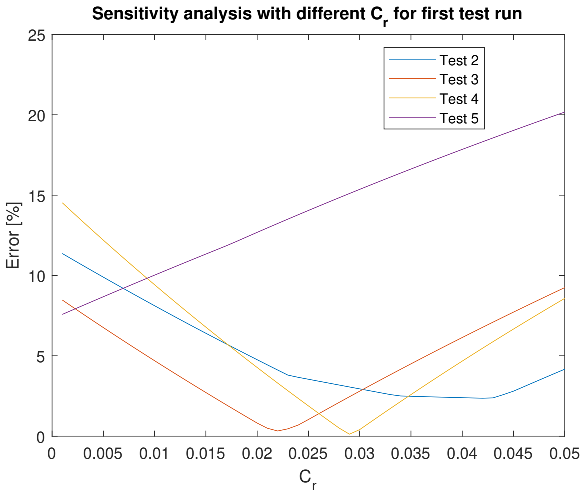


Figure 4.13: *Sensitivity analysis with different C_r using MFF-RLS for the tests during the first test run $\lambda_1 = 1$ and $\lambda_2 = 0.99$.*

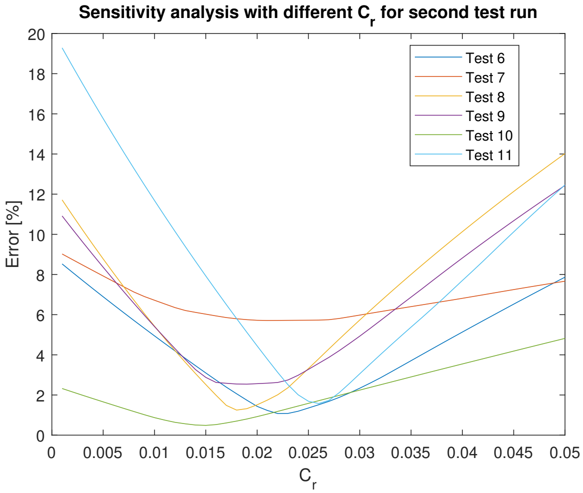


Figure 4.14: *Sensitivity analysis with different C_r using MFF-RLS for the tests during the second test run using $\lambda_1 = 0.999$ and $\lambda_2 = 0.99$.*

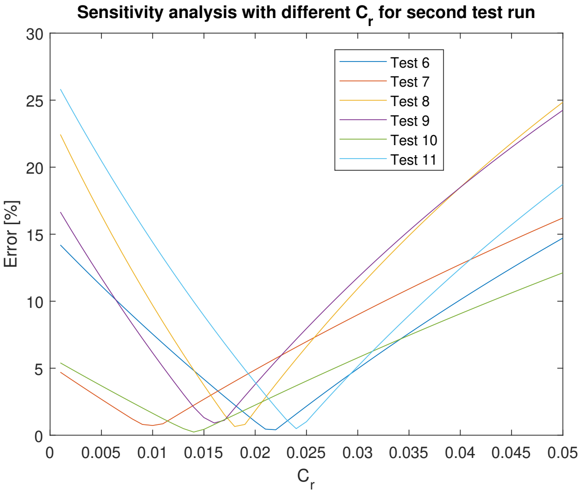


Figure 4.15: *Sensitivity analysis with different C_r using MFF-RLS for the tests during the second test run using $\lambda_1 = 1$ and $\lambda_2 = 0.99$.*

Discussion

From Figure 4.10 and Figure 4.11 it can be seen that the SFF-RLS is robust to changes in the rolling resistance coefficient. The mean error changes very little when the coefficient changes, even up to $C_r = 0.05$. As mentioned in Section 4.3.1, the grade estimate is near-constant and near zero, and by examining the vehicle model, (2.4), it can be seen that C_r depends on the road grade. Given this, the SFF-RLS is not that sensitive to changes in C_r but it could become an issue if the estimated road grade is far from the true road grade.

The MFF-RLS is more sensitive to changes in C_r, but it also gives a very small mean error if the coefficient is chosen correctly. When using $\lambda_1 = 1$ the algorithm is more sensitive to changes but the minimum error can become very close to 0 %. On the other hand, if the coefficient is estimated or chosen poorly the error of the estimate can become very large. A change of 0.01 in C_r can result in a mean error that is 10 percentage points larger. As C_r typically changes between 0-0.3, this indicates that we need to have some knowledge of the road conditions, if it is snowy conditions, dry conditions, or rainy conditions. The C_r could change about 0.01 between snow conditions and dry conditions. There are several companies and car manufacturers that estimate the friction in real-time which could make the estimate of C_r good. When using $\lambda_1 = 0.999$ the algorithm becomes more robust to changes in C_r but not as robust as SFF-RLS.

During the first test run C_r was estimated to 0.01, but the value of C_r that gives the minimum error is between 0.2 and 0.3, for all three algorithms. One reason could be that the road conditions could be different from the road where the coast-down test was performed, which makes the estimation of C_r too low. The difference in road conditions could be due to the road being de-iced between the coast-down test and the other tests during the first test run.

For test 11 the value of C_r that gives the minimum error is larger than the other tests for both sets of forgetting factors for MFF-RLS. Test 11 is highway driving with 400 kg extra load in the car and the higher value of optimal C_r could be because during higher speeds other forces will affect the vehicle dynamics. If other forces affect the vehicle the rolling resistance coefficient will compensate for this and a higher value of C_r will give the smallest error. The rolling resistance could also depend on the speed, but the other highway test, test 10, does not imply that the speed by itself has that much effect as the optimal value for test 10 is around $C_r = 0.02$ for both versions of MFF-RLS. Given that the MFF-RLS is fast another cause could be wheel slip. If slip is present the longitudinal model is not valid, and as presented in Section 4.3.2 the estimate changes very little after the initial 15 seconds. If the wheels slip during this period the estimate could become far from the true mass. If using MFF-RLS with $\lambda_1 = 0.999$ the estimate would slowly converge towards the true mass but if the initial estimate is far from the true mass it might take a long time before giving a good estimate. When using MFF-RLS with $\lambda_1 = 1$ the estimate would not converge towards the true mass. This could be a reason why test 11 gives a larger error for $C_r = 0.02$ than the other tests.

4.4.2 Drag coefficient

The sensitivity of the drag coefficient C_{sd} was examined by keeping all other parameters constant and changing C_{sd}. The coefficient typically varies between 0.4 – 1.5 for different cars, and therefore C_{sd} was varied between 0 – 2 [19]. Note that the estimated value is $C_{sd} = 1.0512$. In Figure 4.16 the mean error using SFF-RLS is shown for all tests. From the figure, it can be seen that C_{sd} needs to change up to $C_{sd} = 1.5$ before the error is above 10 % for all tests but test 11. For test 11 the error is within 10 % up to $C_{sd} = 1.32$.

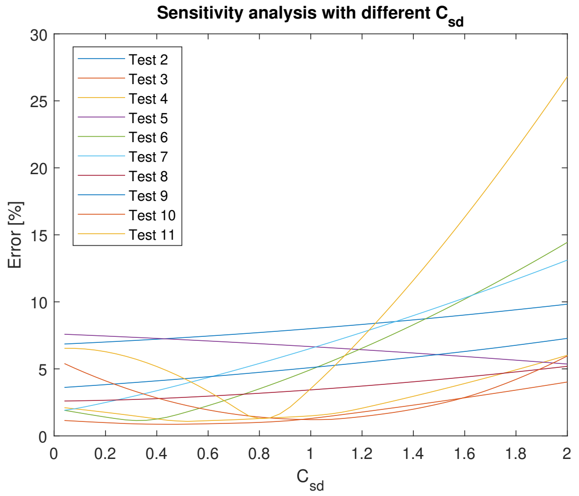


Figure 4.16: *Sensitivity analysis with different C_{sd} using SFF-RLS with $\lambda = 1$.*

In Figure 4.17, the mean error when using MFF-RLS with $\lambda = 0.999$ is shown. MFF-RLS with $\lambda_1 = 1$ is shown in Figure 4.18. For the first set of λ_1 the estimate changes only a few percentage points between the smallest value of C_{sd} and the largest for all tests but test 7. For test 7 C_{sd} needs to change above 1.5 before the error becomes larger than 10 %. By examining Figure 4.18, it can be seen that when using $\lambda_1 = 1$ the algorithm is robust to changes in C_{sd} as the error only changes 3-4 percentage points for the whole span of C_{sd}.

Discussion

The surface area and drag coefficient have a small influence on the estimate both for the SFF-RLS and MFF-RLS. The mean error of the estimate only changes a few percentage points for the whole span of C_{sd}. The MFF-RLS is more robust to changes in this parameter and the error only changes a few percentage points, but for the SFF-RLS the error increases when C_{sd} increases. The C_{sd} needs to increase above 1.5 before the error starts to increase, and due to the span that the coefficient typically is within, the algorithm can be seen as robust to changes in the parameter.

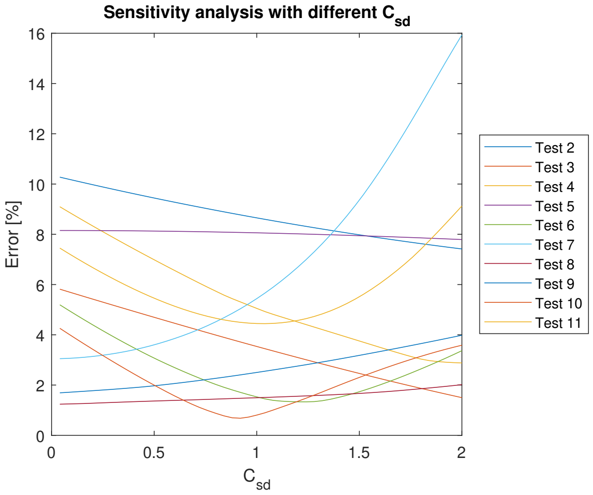


Figure 4.17: *Sensitivity analysis with different C_{sd} using MFF-RLS with $\lambda_1 = 0.999$ and $\lambda_2 = 0.99$.*

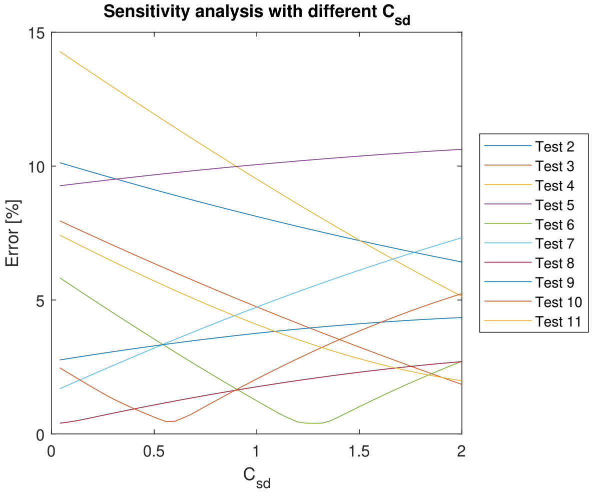


Figure 4.18: *Sensitivity analysis with different C_{sd} using MFF-RLS with $\lambda_1 = 1$ and $\lambda_2 = 0.99$.*

If a roof box is attached to the roof of the car the drag area increases. The total drag could increase around 25-30 % [2]. In our case, this would correspond to C_{sd} increasing up to around 1.37. The error would start to increase for the SFF-RLS and the error would increase for the MFF-RLS but not as much as for the SFF-RLS. Given this, the error would only increase a few percentage points if a roof box is attached to the vehicle. If there was a possibility to adjust C_{sd} in the algorithm when a roof box is attached the estimate would become better.

4.4.3 Wheel radius

To investigate the sensitivity of the wheel radius it was varied and the mass was estimated, both for SFF-RLS and MFF-RLS. The sensitivity for all tests when using SFF-RLS is shown in Figure 4.19. When altering r_{wheel} around 2 centimeters the error changes around 5 percentage points. The wheel radius which gives the smallest mean error varies from 0.32 – 0.39 m, compared to the measured wheel radius 0.358 m. Most of the tests give the smallest error for r_{wheel} around 0.36 – 0.37 m.

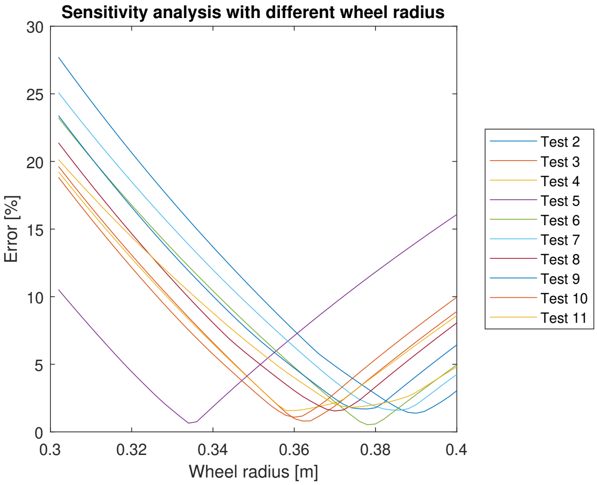


Figure 4.19: *Sensitivity analysis with different wheel radius using SFF-RLS with $\lambda = 1$.*

The mean error for all tests using MFF-RLS is shown in Figure 4.20 and 4.21 for $\lambda_1 = 0.999$ and $\lambda_1 = 1$ respectively. When looking at the error when using MFF-RLS the range of the value that gives the smallest mean error varies from 0.32 – 0.39 m for both sets of λ_1. When using $\lambda_1 = 1$ the value that gives the smallest error is more spread out than for $\lambda_1 = 0.999$. For both sets of λ_1 the radius can change ± 4 cm from the optimal value and still be within 10 % error. When comparing the SFF-RLS and MFF-RLS we see that the value that gives the smallest mean error varies a bit for each test for MFF-RLS using $\lambda_1 = 1$, where it is mostly the same value for most tests for SFF-RLS and MFF-RLS with $\lambda_1 = 0.999$.

Discussion

When changing the wheel radius the estimate changes a bit but the radius needs to have a fairly large change, about 5 cm before the error is above 10 %. For SFF-RLS and MFF-RLS with $\lambda_1 = 0.999$ most tests give about the same value

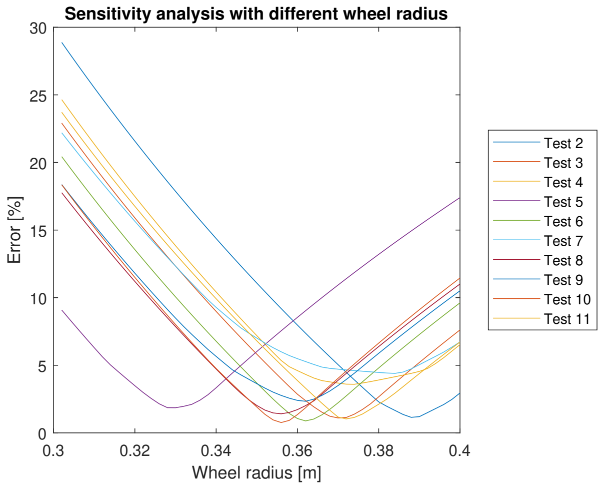


Figure 4.20: *Sensitivity analysis with different wheel radius using MFF-RLS with $\lambda_1 = 0.999$ and $\lambda_2 = 0.99$.*

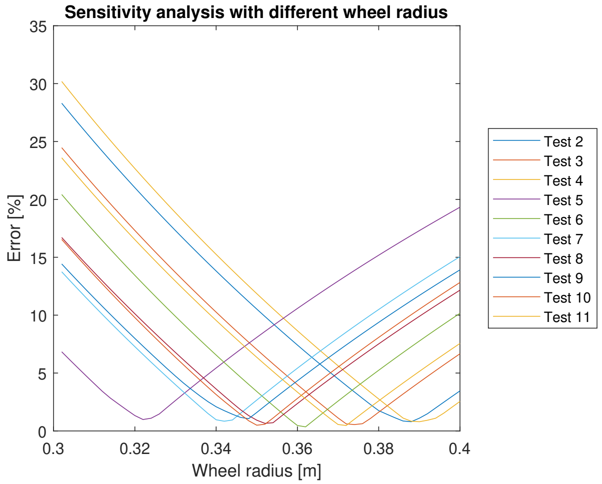


Figure 4.21: *Sensitivity analysis with different wheel radius using MFF-RLS with $\lambda_1 = 1$ and $\lambda_2 = 0.99$.*

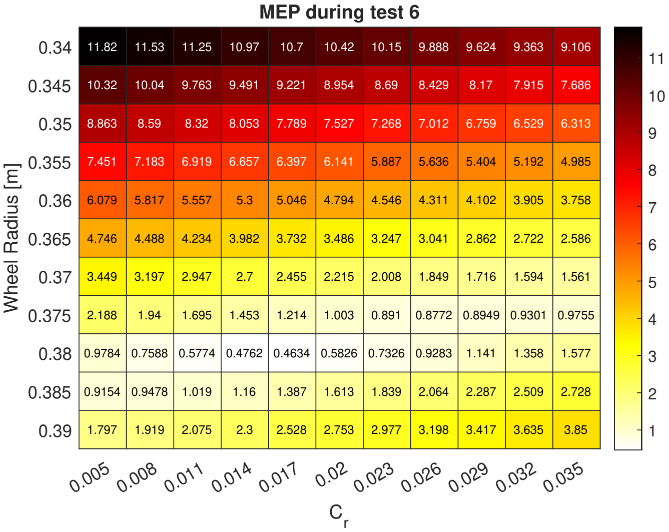
for the wheel radius where the mean error is minimized, but for MFF-RLS with $\lambda_1 = 1$ the value is more spread out between the tests. The value of r_{wheel} is slightly larger than the value used for most tests, which could indicate that the measured value is too small. It could also indicate that the wheel torque is slightly overestimated, as a larger wheel radius gives a smaller estimated wheel force, according to (3.1).

Test 5 gives a minimum error for a value of $r_{\text{wheel}} = 0.33$ which is lower than for the other tests. As discussed in Section 4.3.2 the mass estimate is far from the true mass after around 48 seconds and then slowly converges towards the true mass. When using a smaller wheel radius the estimate is closer to the true value after 50 seconds, which makes the error smaller. If discarding the first 50 seconds of test 5, the value of r_{wheel} that minimizes the error is around $r_{\text{wheel}} = 0.36$.

If the type of the wheel changes the radius can change a few centimeters and therefore it is necessary for the algorithm to know the radius of the wheel with good accuracy, within 3-4 centimeters, in order to give a good estimate.

4.4.4 Combined sensitivity

The sensitivity with error in multiple parameters was examined by varying the wheel radius and rolling resistance. The drag coefficient can be estimated one time for each model and is constant for all runs. This combined with the indication that all algorithms are robust to changes in drag coefficient, only the wheel radius and rolling resistance are examined. The wheel radius and rolling resistance were varied and the MEP for each test was calculated. The MEP for all tests, for all three algorithms, can be found in Appendix C. The sensitivity for test 6 when using SFF-RLS is shown in Figure 4.22. From the figure, it can be seen that the MEP does not change much when C_r changes. The MEP varies more with r_{wheel}, with a large MEP if the r_{wheel} is small. The MEP is minimized when $r_{\text{wheel}} = 0.38$ which corresponds to the results in Section 4.4.3.

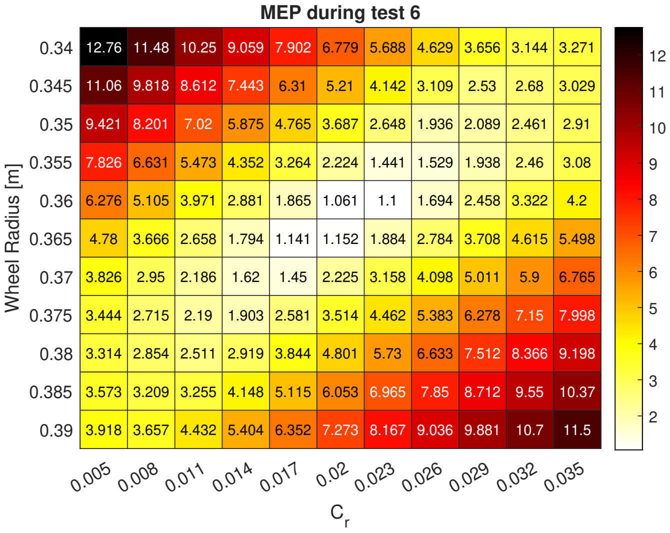


	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	11.82	11.53	11.25	10.97	10.7	10.42	10.15	9.888	9.624	9.363	9.106
0.345	10.32	10.04	9.763	9.491	9.221	8.954	8.69	8.429	8.17	7.915	7.686
0.35	8.863	8.59	8.32	8.053	7.789	7.527	7.268	7.012	6.759	6.529	6.313
0.355	7.451	7.183	6.919	6.657	6.397	6.141	5.887	5.636	5.404	5.192	4.985
0.36	6.079	5.817	5.557	5.3	5.046	4.794	4.546	4.311	4.102	3.905	3.758
0.365	4.746	4.488	4.234	3.982	3.732	3.486	3.247	3.041	2.862	2.722	2.586
0.37	3.449	3.197	2.947	2.7	2.455	2.215	2.008	1.849	1.716	1.594	1.561
0.375	2.188	1.94	1.695	1.453	1.214	1.003	0.891	0.8772	0.8949	0.9301	0.9755
0.38	0.9784	0.7588	0.5774	0.4762	0.4634	0.5826	0.7326	0.9283	1.141	1.358	1.577
0.385	0.9154	0.9478	1.019	1.16	1.387	1.613	1.839	2.064	2.287	2.509	2.728
0.39	1.797	1.919	2.075	2.3	2.528	2.753	2.977	3.198	3.417	3.635	3.85

Figure 4.22: *Sensitivity analysis during test 6 with different C_r and r_{wheel} using SFF-RLS with $\lambda = 1$.*

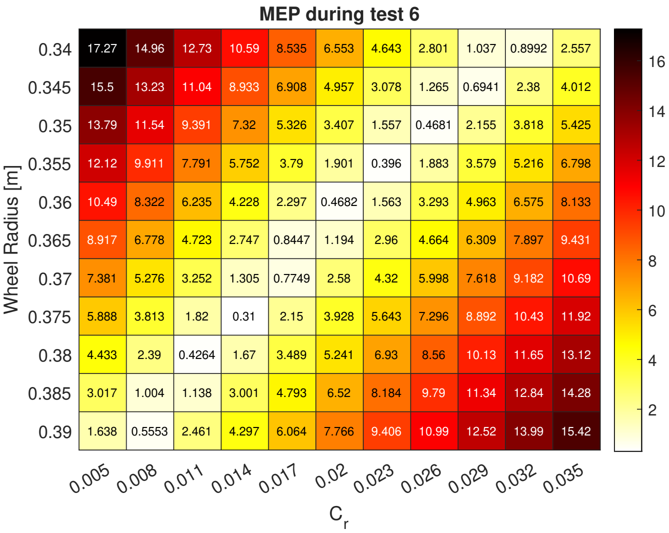
The sensitivity for the MFF-RLS with $\lambda_1 = 0.999$, $\lambda_2 = 0.99$ can be seen in Figure 4.23, and the MFF-RLS with $\lambda_1 = 1$, $\lambda_2 = 0.99$ is shown in Figure 4.24.

From the figures it can be seen that the MEP becomes large if both C_r and r_{wheel} is large or if both are small. The MEP becomes smaller if the values are chosen on the diagonal, with small C_r and large r_{wheel} or large C_r and small r_{wheel}. The used value of C_r is $C_r = 0.02$, and if the C_r is within 0.02 ± 0.005 the wheel radius needs to be between $0.35 \leq r_{\text{wheel}} \leq 0.38$ in order for the MEP to be smaller than 8 %, for both sets of λ_1.



Wheel Radius [m]	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	12.76	11.48	10.25	9.059	7.902	6.779	5.688	4.629	3.656	3.144	3.271
0.345	11.06	9.818	8.612	7.443	6.31	5.21	4.142	3.109	2.53	2.68	3.029
0.35	9.421	8.201	7.02	5.875	4.765	3.687	2.648	1.936	2.089	2.461	2.91
0.355	7.826	6.631	5.473	4.352	3.264	2.224	1.441	1.529	1.938	2.46	3.08
0.36	6.276	5.105	3.971	2.881	1.865	1.061	1.1	1.694	2.458	3.322	4.2
0.365	4.78	3.666	2.658	1.794	1.141	1.152	1.884	2.784	3.708	4.615	5.498
0.37	3.826	2.95	2.186	1.62	1.45	2.225	3.158	4.098	5.011	5.9	6.765
0.375	3.444	2.715	2.19	1.903	2.581	3.514	4.462	5.383	6.278	7.15	7.998
0.38	3.314	2.854	2.511	2.919	3.844	4.801	5.73	6.633	7.512	8.366	9.198
0.385	3.573	3.209	3.255	4.148	5.115	6.053	6.965	7.85	8.712	9.55	10.37
0.39	3.918	3.657	4.432	5.404	6.352	7.273	8.167	9.036	9.881	10.7	11.5

Figure 4.23: *Sensitivity analysis during test 6 with different C_r and r_{wheel} using MFF-RLS with $\lambda_1 = 0.999$ and $\lambda_2 = 0.99$.*



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	17.27	14.96	12.73	10.59	8.535	6.553	4.643	2.801	1.037	0.8992	2.557
0.345	15.5	13.23	11.04	8.933	6.908	4.957	3.078	1.265	0.6941	2.38	4.012
0.35	13.79	11.54	9.391	7.32	5.326	3.407	1.557	0.4681	2.155	3.818	5.425
0.355	12.12	9.911	7.791	5.752	3.79	1.901	0.396	1.883	3.579	5.216	6.798
0.36	10.49	8.322	6.235	4.228	2.297	0.4682	1.563	3.293	4.963	6.575	8.133
0.365	8.917	6.778	4.723	2.747	0.8447	1.194	2.96	4.664	6.309	7.897	9.431
0.37	7.381	5.276	3.252	1.305	0.7749	2.58	4.32	5.998	7.618	9.182	10.69
0.375	5.888	3.813	1.82	0.31	2.15	3.928	5.643	7.296	8.892	10.43	11.92
0.38	4.433	2.39	0.4264	1.67	3.489	5.241	6.93	8.56	10.13	11.65	13.12
0.385	3.017	1.004	1.138	3.001	4.793	6.52	8.184	9.79	11.34	12.84	14.28
0.39	1.638	0.5553	2.461	4.297	6.064	7.766	9.406	10.99	12.52	13.99	15.42

Figure 4.24: *Sensitivity analysis during test 6 with different C_r and r_{wheel} using MFF-RLS with $\lambda_1 = 1$ and $\lambda_2 = 0.99$.*

Discussion

For the SFF-RLS it can be seen that C_r has a small effect on the MEP, and r_{wheel} affects the MEP the most. This is because when using SFF-RLS with $\lambda = 1$ the road grade estimate is close to 0 degrees, and this was further discussed in Section 4.4.1.

For both the MFF-RLS algorithms it can be seen that if C_r is estimated too low the error becomes smaller if r_{wheel} becomes larger. This is reasonable when looking at the rolling resistance force (2.4) and the wheel force (3.1). When C_r is lowered the rolling resistance force becomes smaller, and when the wheel radius increases the wheel force becomes smaller. When combining error in both C_r and r_{wheel} it can be seen that unless the error is large in both parameters at the same time, the MFF-RLS algorithms give a reasonable estimate. The C_r needs to be known with an accuracy of around $0.005 - 0.01$ and the wheel radius within 2-3 centimeters. This is in line with the results shown in the previous sections.

4.4.5 Summarizing discussion

In summary, the MFF-RLS gives an estimate more quickly than SFF-RLS, and the estimate is fairly constant. When using MFF-RLS with $\lambda_1 = 1$ the estimate is quicker than when using $\lambda_1 = 0.999$. Both versions of the MFF-RLS are more sensitive to changes in C_r, but when choosing it optimally the error becomes very small. The SFF-RLS is more robust to changes in the rolling resistance and drag coefficient, and error in the wheel radius affects MFF-RLS and SFF-RLS about the same. When estimating all parameters correctly the MFF-RLS has the potential of estimating the mass with very little error, but the SFF-RLS gives a good estimate in all different tests even though some parameters might be estimated with less accuracy.

The SFF-RLS with a forgetting factor $\lambda = 1$ gives a road grade estimate that is very low, which could affect the mass estimate negatively if driving on a big slope. Another aspect that needs to be taken into consideration is that the timing of the slope also affects the estimate, where the estimate will be more affected if the slope is at the beginning of the run compared to after a long time of driving. The MFF-RLS has a forgetting factor that is less than 1 for road grade and therefore does not have this problem.

The mass is near-constant, and only changes around 2 kg for 30 minutes of driving, and therefore a realistic value for the forgetting factor for the mass is very close to 1. If using $\lambda_1 = 0.999$ a value near 1 is used, but the algorithm is able to converge towards the true mass even if the mass estimate is far from the true value. The road grade varies more during a run, therefore a lower value of forgetting factor is reasonable to use. When using $\lambda_2 = 0.99$ approximately 100 samples are used, and this is a more realistic value to use than using all samples for the road grade. This would in theory also improve the results if driving on steep slopes or if the road grade varies much.

All three algorithms are viable and when choosing between them the different characteristics decide what algorithm to choose. Both the MFF-RLS algorithms are very quick, and if the parameters are estimated well the algorithm with $\lambda_1 = 1$ is to prefer. That algorithm also gives a very stable estimate that does not change that much during the run. The SFF-RLS is more robust to errors in the estimates of the parameters.

As described in Section 3.1 the algorithm uses wheel torque that is calculated in the electronic control unit, ECU, in the engine. Algorithms proposed in the literature estimate the dynamics in the driveline in order to estimate the wheel torque. The results indicate that the estimation of wheel torque in the ECU is good and works well in the algorithm, but does not give any significant performance improvements.

When comparing the algorithm to the ones from related work it can be seen that the algorithm performs in line with most from the literature. The SFF-RLS has an average error of 4.42 % and MFF-RLS has an average error of 4.15 % when using $\lambda_1 = 0.999$ and 4.97 % when using $\lambda_1 = 1$. Ghosh et al. [6] achieve 5 % error when using an RLS estimator with a torque observer, Lin et al. [11] get an error of 9 % on a heavy-duty vehicle and Jonsson Holm [8] get an error of 5 % when using EKF. The algorithm presented in this thesis is performing in line or better, and is shown to work in all different driving scenarios.

5

Conclusions and future work

This thesis aimed to develop an estimator and answering questions about its performance. The aim was to develop a software-based estimator and compare different driving scenarios and how they affect the performance of the algorithm. This chapter presents conclusions and a note of future work that can be done on the subject.

5.1 Conclusions

In Section 1.3 four questions were proposed, which were how to adapt the algorithm to wheel torque, how the developed algorithm performs, which driving scenarios have the largest effect on the performance, and which parameters influence the algorithm.

The algorithm and the vehicle model contained the wheel force and as a consequence, it was straightforward to convert the wheel torque to wheel force when the wheel radius is known. The developed algorithm uses the wheel torque calculated in the ECU in the engine as opposed to previous work that estimates the dynamics in the driveline. The results indicate that the estimate from the engine ECU is good and is possible to use in the developed algorithm.

The algorithm performed in the range of previous literature, for both SFF-RLS and MFF-RLS. The average error for SFF-RLS was 4.42 % and for MFF-RLS it was 4.15 % and 4.97 % for $\lambda_1 = 0.999$ and $\lambda_1 = 1$ respectively. Previous literature on the subject has achieved around 5-10 % error and the developed algorithm was slightly better than algorithms in the literature on average. The maximum error for the SFF-RLS was 8.08 %, for MFF-RLS with $\lambda_1 = 0.999$ it was 8.58 % and for MFF-RLS with $\lambda_1 = 1$ the maximum error was 10.09 % which still is on par with several algorithms from the literature.

The different driving scenarios have little effect on the performance as the algorithm performed equally well for the different scenarios, but to draw definitive conclusions more tests would be needed. Highway driving gave the smallest average error but with fewer tests on highways, there were not enough tests to conclude that the algorithm was better for highway driving.

The parameter that affects the algorithms the most was the rolling resistance. The SFF-RLS was robust to changes in all parameters that are tested, including the rolling resistance. The drag coefficient did not have a large effect on the estimate. The wheel radius needs to be known with an accuracy of $3 - 4$ cm, otherwise, the error can become very large, for both SFF-RLS and MFF-RLS.

The drag coefficient does not change and would be possible to estimate one time for each car model with great accuracy. One way to alter the drag coefficient and surface area is if a roof box is attached to the roof of the car, which would increase the drag area. The increased drag area would increase the error by a few percentage points, but if being able to adjust the coefficient the error could be reduced. The rolling resistance changes between every ride and changes throughout the whole ride, which makes it hard to estimate. If being able to estimate the rolling resistance coefficient with good accuracy the estimate would be improved. There are several companies and car manufacturers that estimate the available friction in real-time which could be used in this application. The wheel radius could also change during the ride but should stay within the accuracy limit.

In conclusion, both SFF-RLS and MFF-RLS work well, and choosing one of these algorithms depends on what characteristics are wanted. MFF-RLS with $\lambda_1 = 1$, $\lambda_2 = 0.99$ is to prefer if the parameter estimate is good as the error can become close to 0 if the parameters are chosen correctly. It also gives an estimate that is fairly constant during the run. The SFF-RLS is to prefer if the parameter estimate is uncertain due to it being robust to rolling resistance and drag area. The MFF-RLS with $\lambda_1 = 0.999$, $\lambda_2 = 0.99$ is to prefer if the parameter estimate is fairly good and a reactive estimate is wanted.

5.2 Future work

Given as the sensitivity analysis shows that for both MFF-RLS algorithms the error can become very low if the rolling resistance is correctly estimated. If implementing this in a car the performance could improve significantly if the rolling resistance could be estimated each time the car was driven.

It would also be interesting to further investigate how the rolling resistance changes with speed. In some literature, it is mentioned that the rolling resistance increases with speed, and therefore it would be interesting to implement a function for estimating the rolling resistance [12].

In future work, one aspect that could be examined is what happens if the mass changes during a run, for example, if one passenger enters or exits the vehicle. What would happen with the mass estimate in this case, how do the different algorithms react?

Another aspect that is interesting to investigate is what happens when the surface changes, for example when driving on asphalt and then driving on a gravel road. What algorithm reacts best to a change of surface?

If this was to be implemented in a real car it would also be interesting to see what happens if the algorithm can be able to save certain parameters and use them when initializing the algorithm with these values the next time it is run. For example, the estimated weight could be used to initialize the algorithm when it is run the next time.

Appendix

A

Appendix A

A.1 Measurement Signals

All signals used from the CAN-bus are listed in table A.1.

Table A.1: *All used signals on the CAN-bus*

Signal	Unit
Time	s
WheelTorque	Nm
Target Gear	-
Current Gear	-
Brake Info Status	-
Parking Brake Status	-
Longitudinal Acceleration	m/s^2
Transversal Acceleration	m/s^2
Wheel Speed Front Right	rpm
Wheel Speed Front Left	rpm
Vehicle Speed	km/h
Wheel Speed Rear Left	rpm
Wheel Speed Rear Right	rpm
Fuel Level	l

B Appendix B

In this appendix plots for all tests are presented.

B.1 Single forgetting factor

In this section the plots for the test are presented, where a single forgetting factor was used. The forgetting factor used was $\lambda = 1$. In each section, the mass estimate and road grade estimate for each test are presented.

B.1.1 Test 2

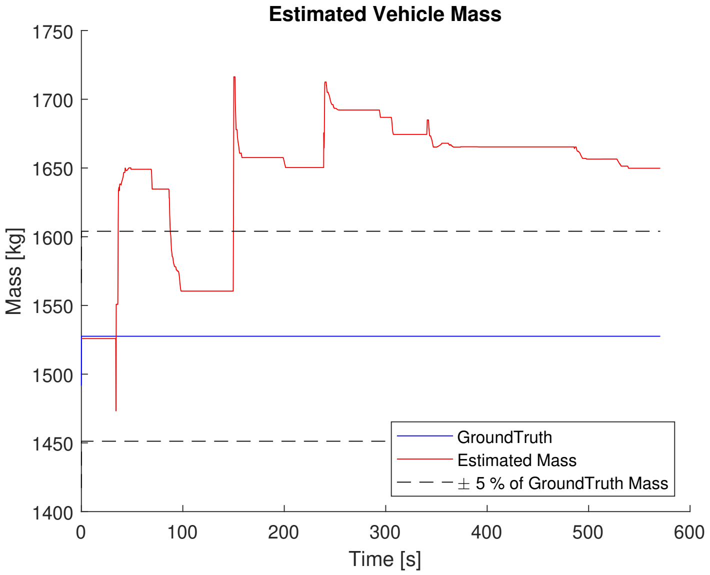


Figure B.1: Mass estimate during test 2 with a single forgetting factor set to 1.

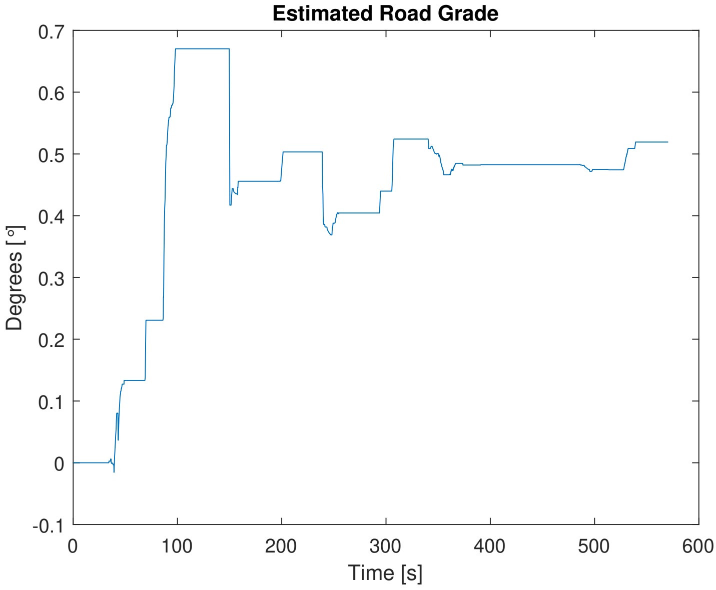


Figure B.2: Road grade estimate during test 2 with a single forgetting factor set to 1.

B.1.2 Test 3

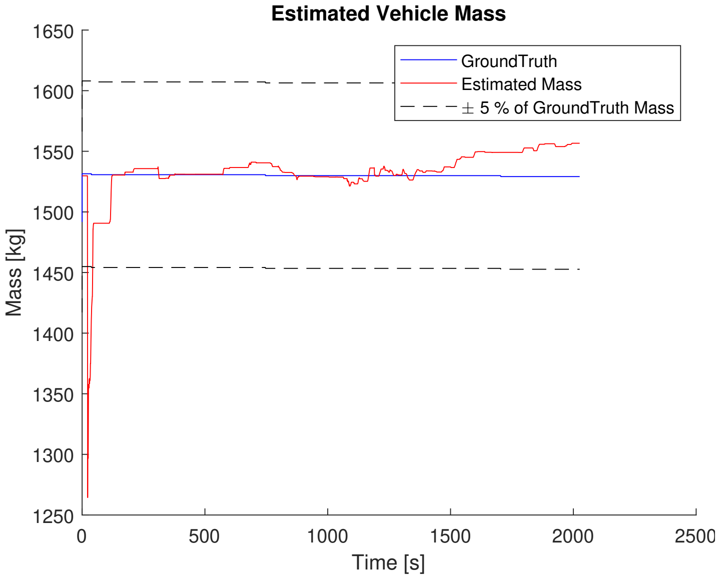


Figure B.3: Mass estimate during test 3 with a single forgetting factor set to 1.

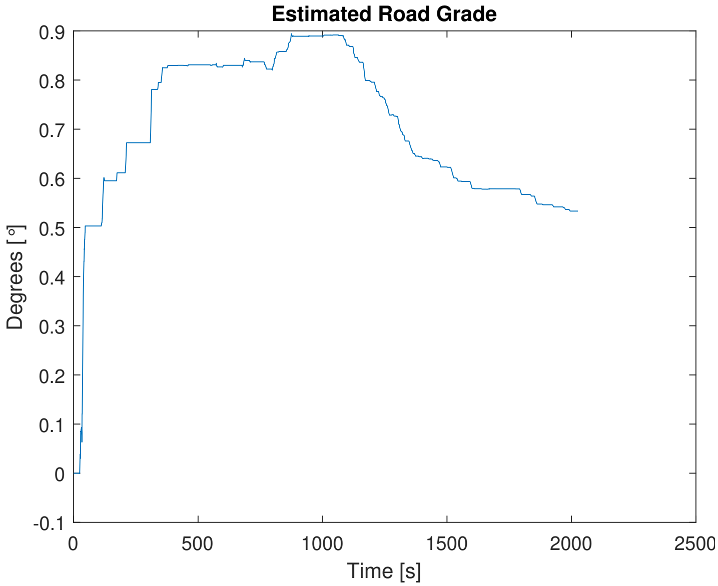


Figure B.4: Road grade estimate during test 3 with a single forgetting factor set to 1.

B.1.3 Test 4

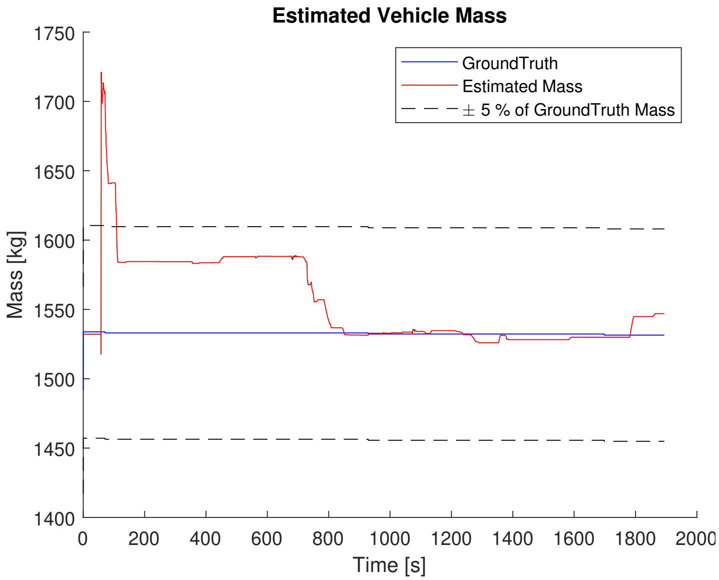


Figure B.5: *Mass estimate during test 4 with a single forgetting factor set to 1.*

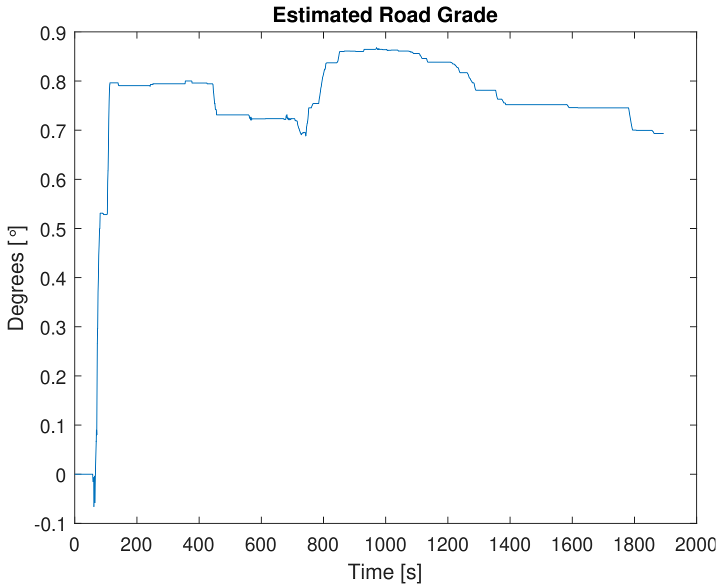


Figure B.6: *Road grade estimate during test 4 with a single forgetting factor set to 1.*

B.1.4 Test 5

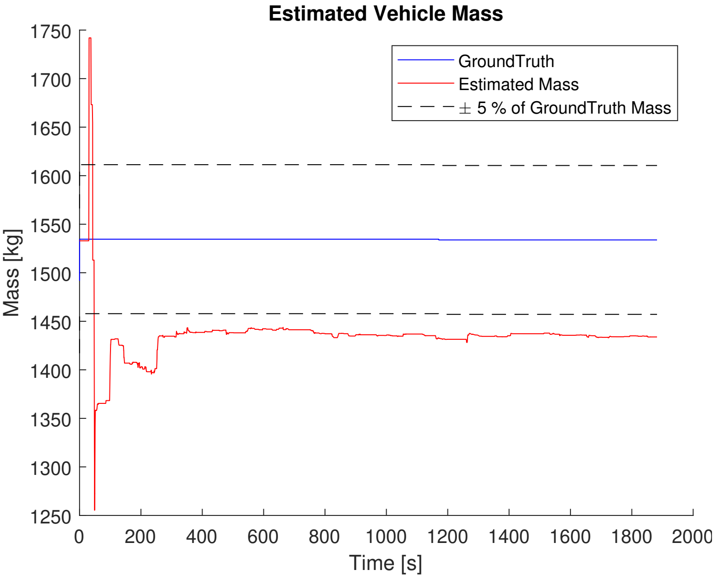


Figure B.7: *Mass estimate during test 5 with a single forgetting factor set to 1.*

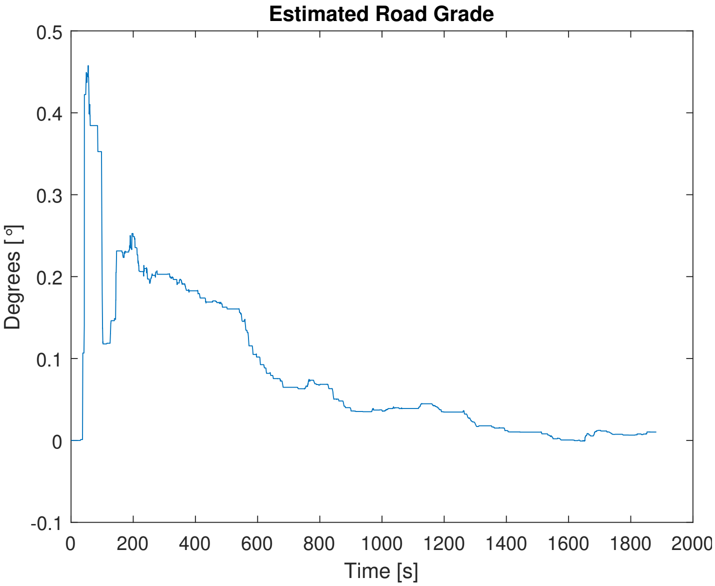


Figure B.8: *Road grade estimate during test 5 with a single forgetting factor set to 1.*

B.1.5 Test 6

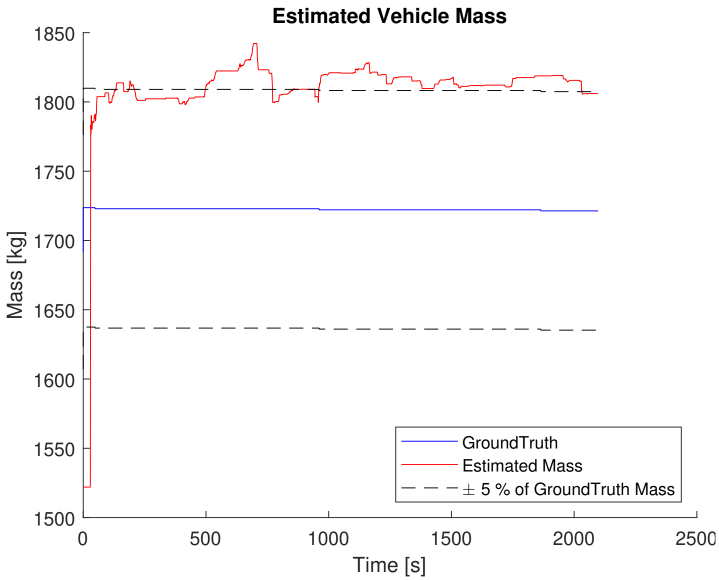


Figure B.9: Mass estimate during test 6 with a single forgetting factor set to 1.

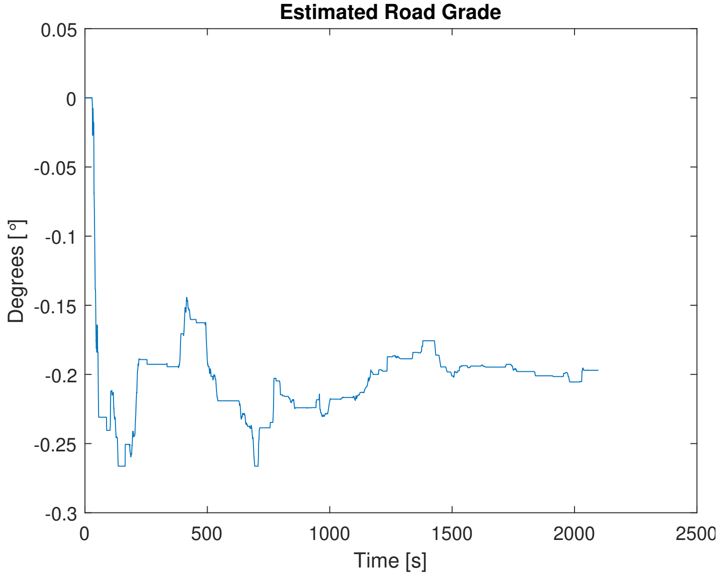


Figure B.10: Road grade estimate during test 6 with a single forgetting factor set to 1.

B.1.6 Test 7

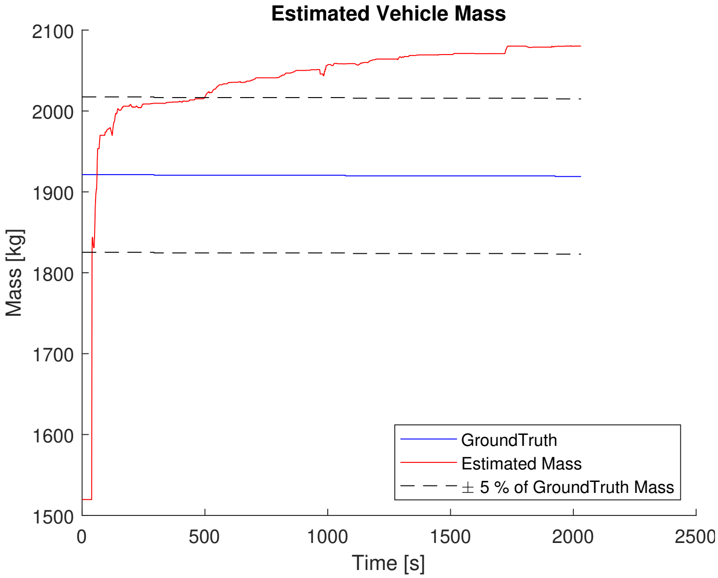


Figure B.11: *Mass estimate during test 7 with a single forgetting factor set to 1.*

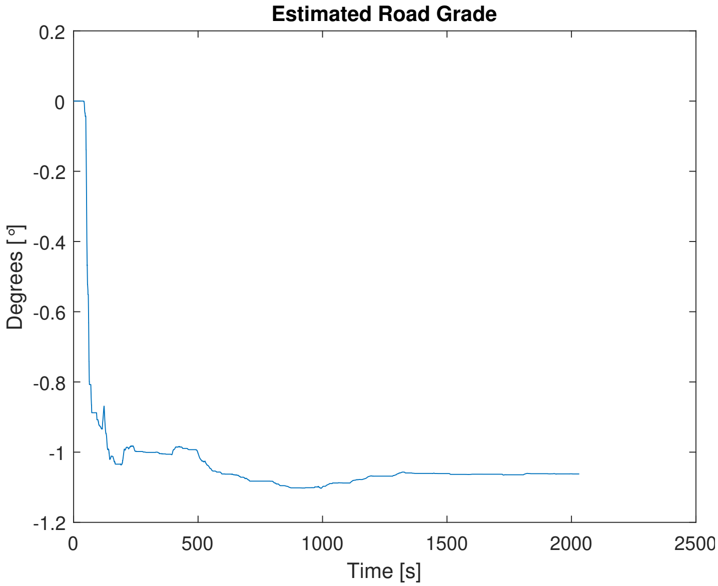


Figure B.12: *Road grade estimate during test 7 with a single forgetting factor set to 1.*

B.1.7 Test 8

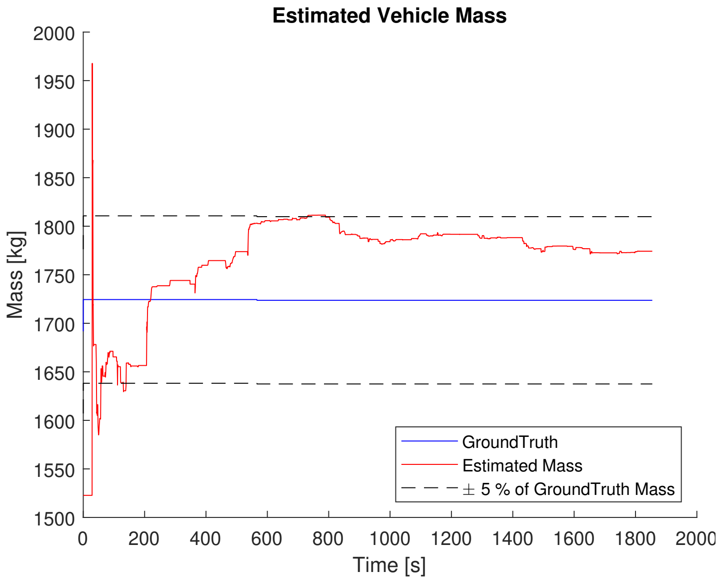


Figure B.13: *Mass estimate during test 8 with a single forgetting factor set to 1.*

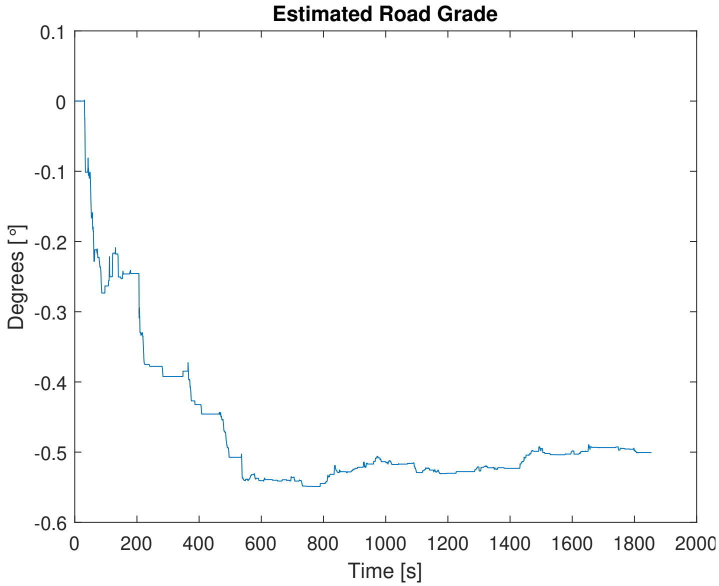


Figure B.14: *Road grade estimate during test 8 with a single forgetting factor set to 1.*

B.1.8 Test 9

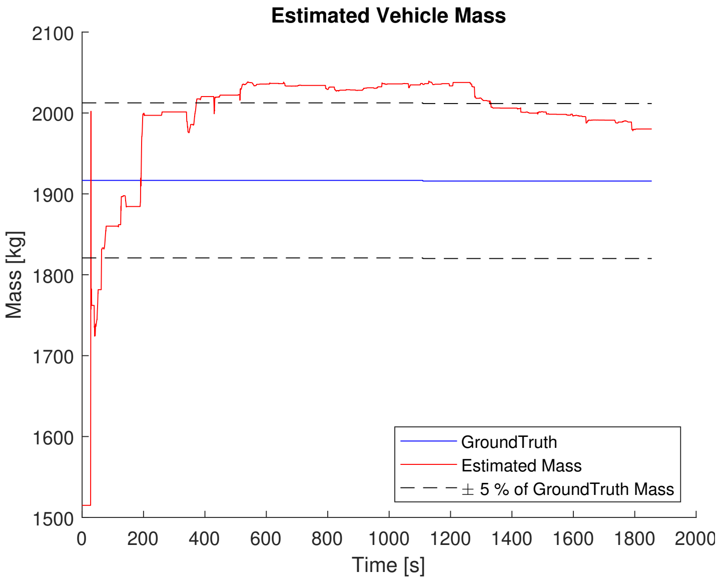


Figure B.15: *Mass estimate during test 9 with a single forgetting factor set to 1.*

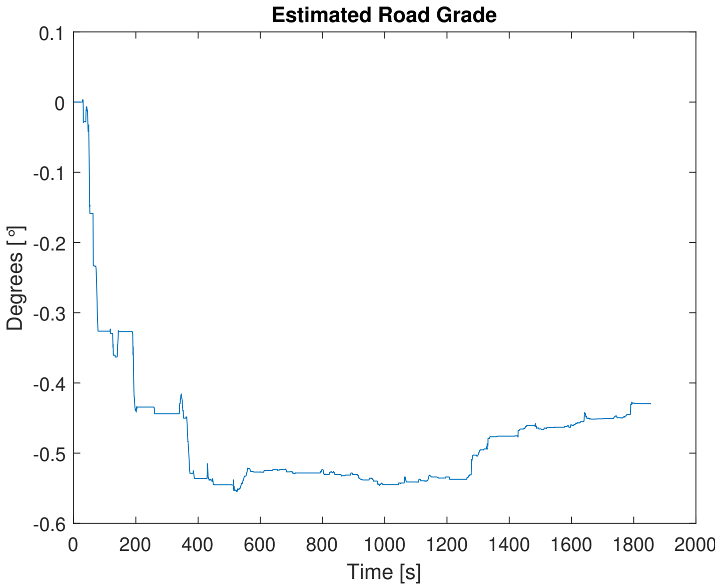


Figure B.16: *Road grade estimate during test 9 with a single forgetting factor set to 1.*

B.1.9 Test 10

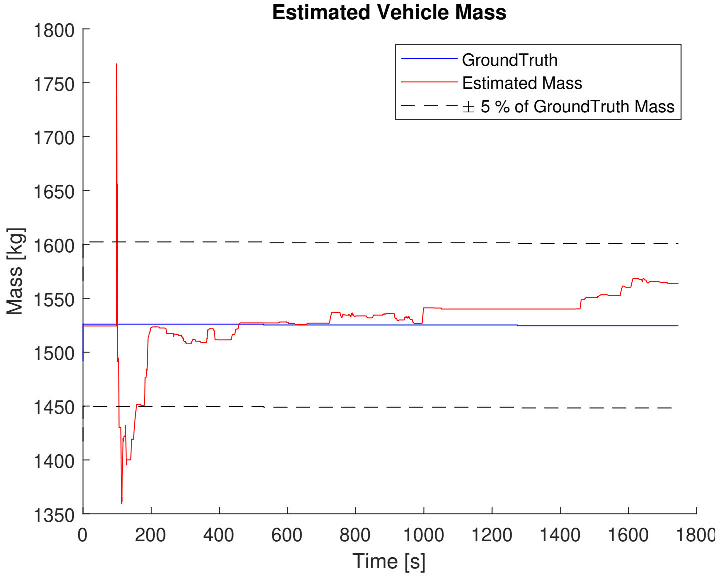


Figure B.17: *Mass estimate during test 10 with a single forgetting factor set to 1.*

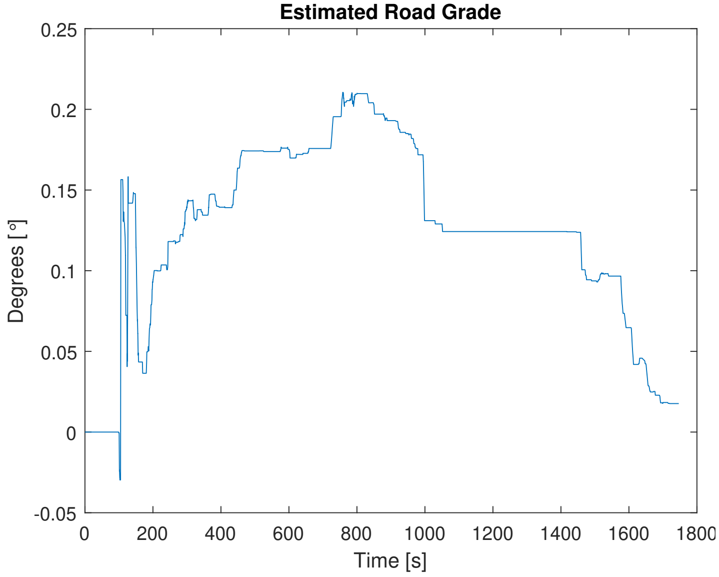


Figure B.18: *Road grade estimate during test 10 with a single forgetting factor set to 1.*

B.1.10 Test 11

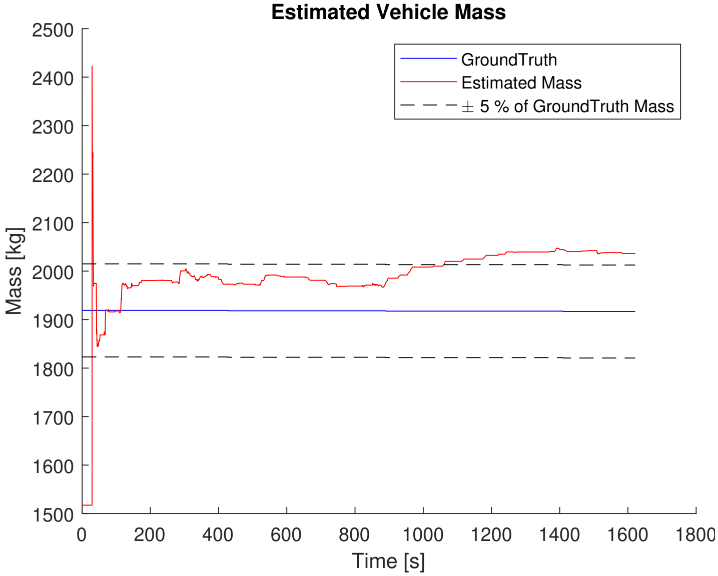


Figure B.19: *Mass estimate during test 11 with a single forgetting factor set to 1.*

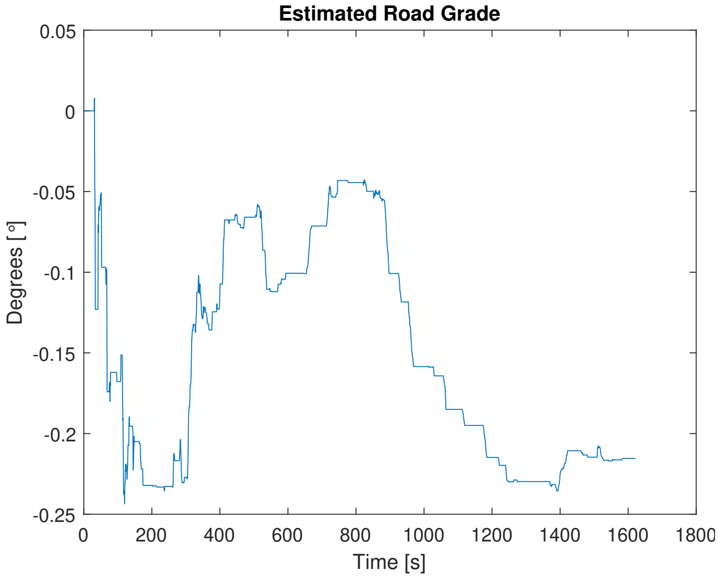


Figure B.20: *Road grade estimate during test 11 with a single forgetting factor set to 1.*

B.2 Multiple forgetting factors

In this section the plots for the test are presented, where multiple forgetting factors were used. Two sets of forgetting factors were used. The two sets used were $\lambda_1 = 0.999$, $\lambda_2 = 0.99$ and $\lambda_1 = 1$, $\lambda_2 = 0.99$. Both sets are presented under each test.

B.2.1 Test 2

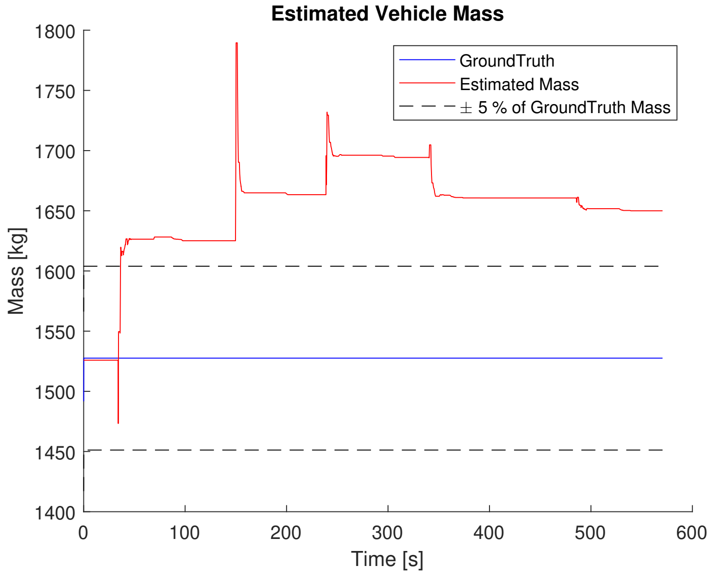


Figure B.21: *Mass estimate during test 2 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

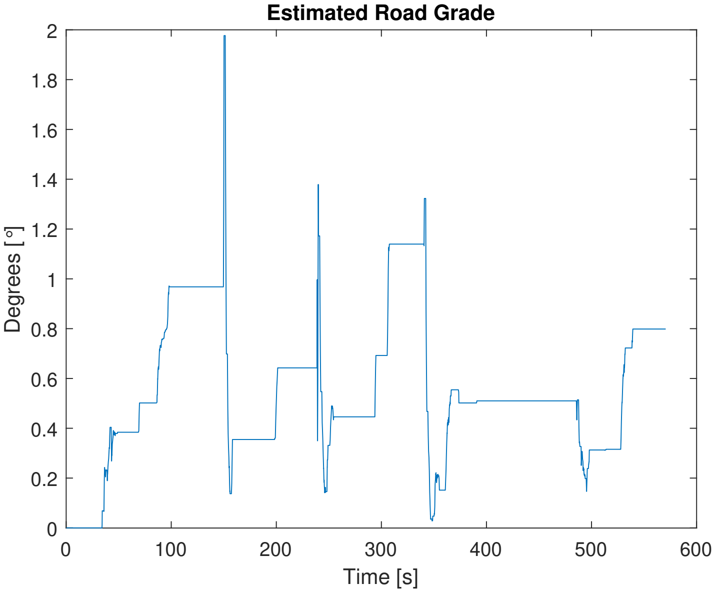


Figure B.22: *Road grade estimate during test 2 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

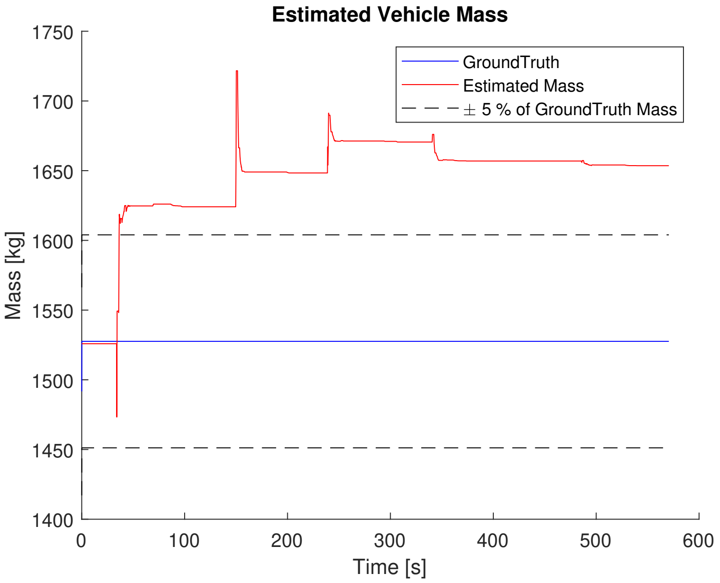


Figure B.23: Mass estimate during test 2 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.

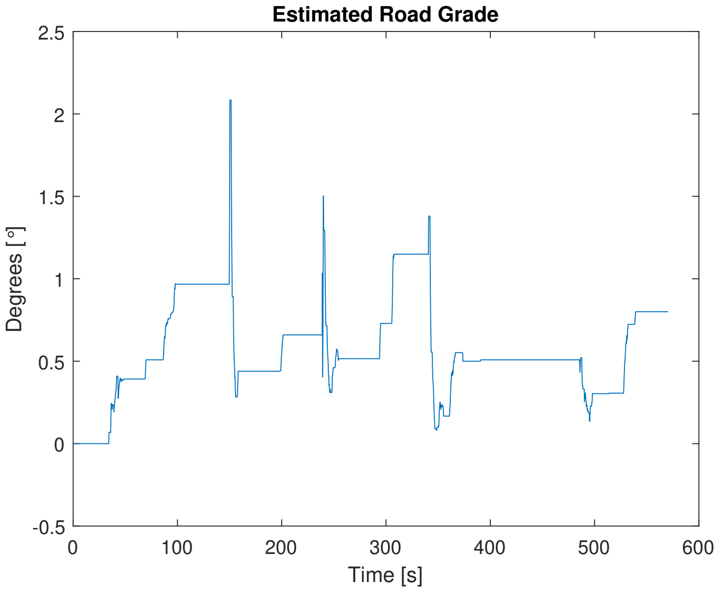


Figure B.24: Road grade estimate during test 2 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.

B.2.2 Test 3

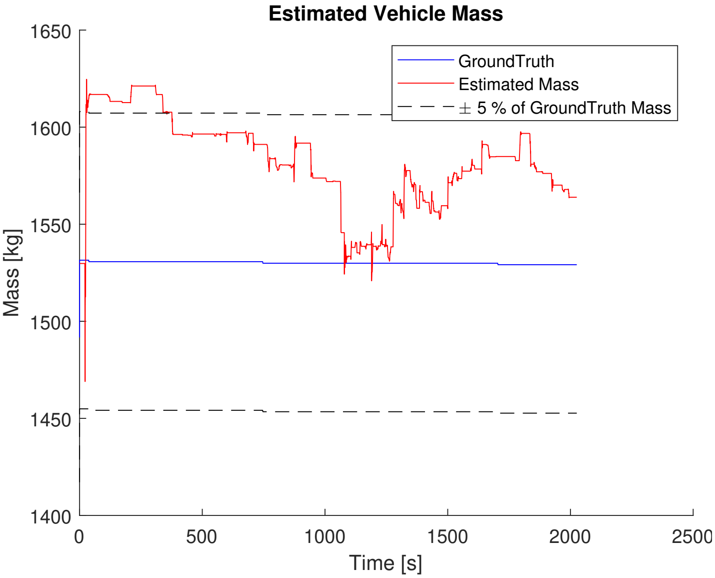


Figure B.25: *Mass estimate during test 3 with multiple forgetting factors* $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.

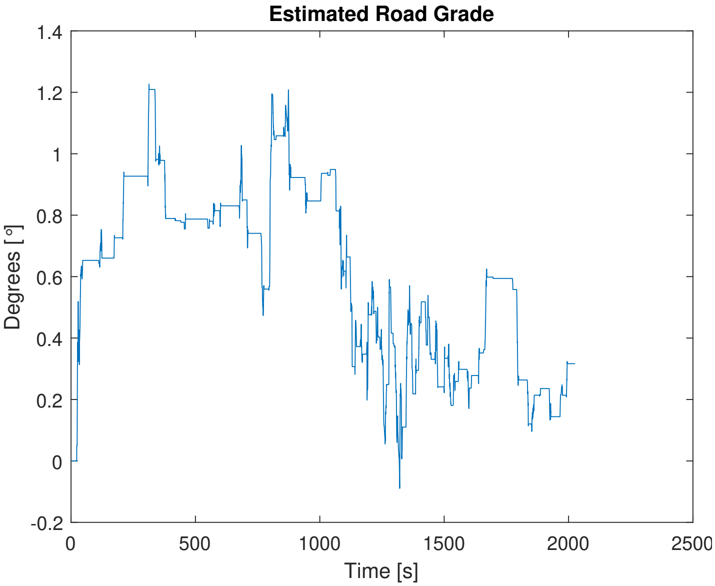


Figure B.26: *Road grade estimate during test 3 with multiple forgetting factors* $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.

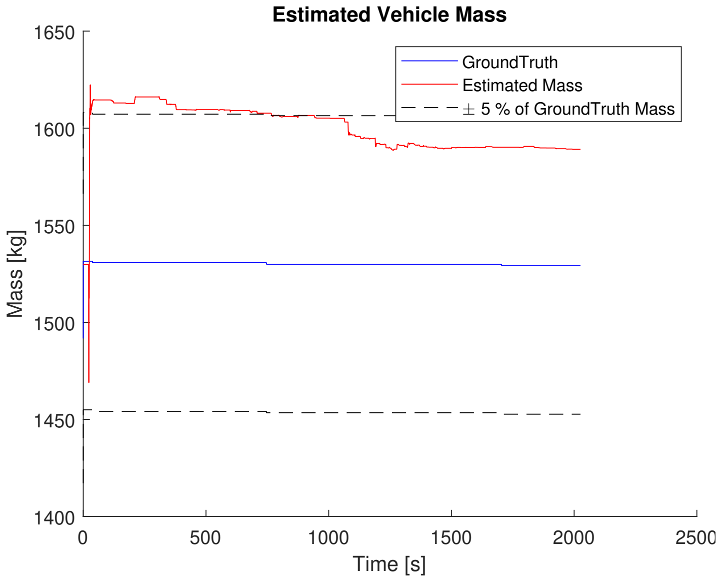


Figure B.27: *Mass estimate during test 3 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

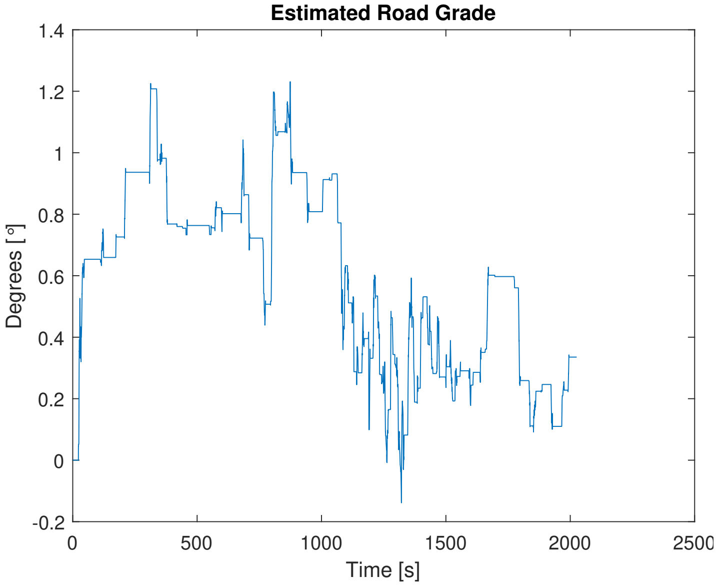


Figure B.28: *Road grade estimate during test 3 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

B.2.3 Test 4

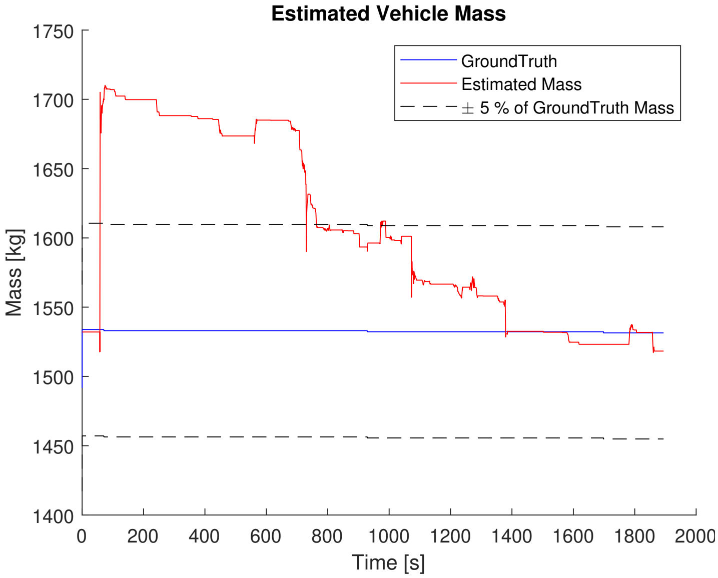


Figure B.29: *Mass estimate during test 4 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

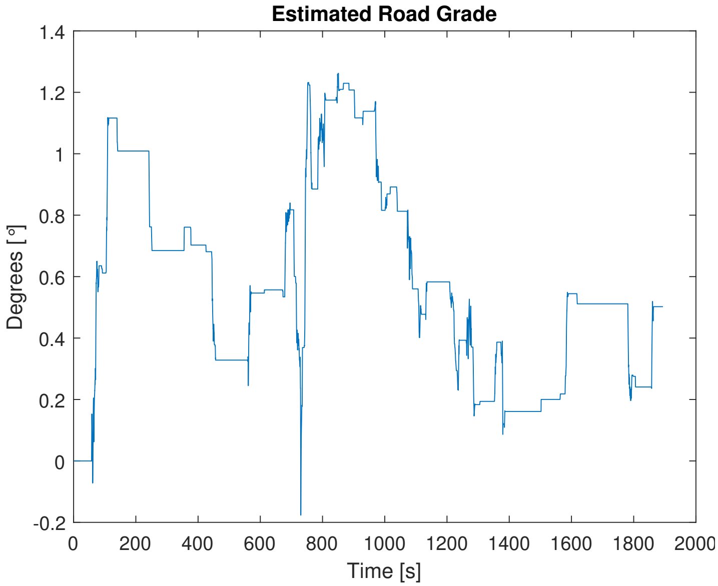


Figure B.30: *Road grade estimate during test 4 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

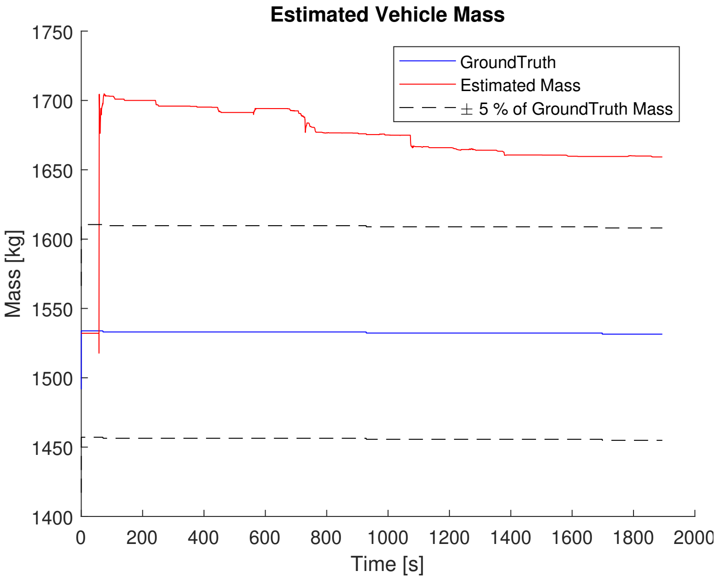


Figure B.31: *Mass estimate during test 4 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

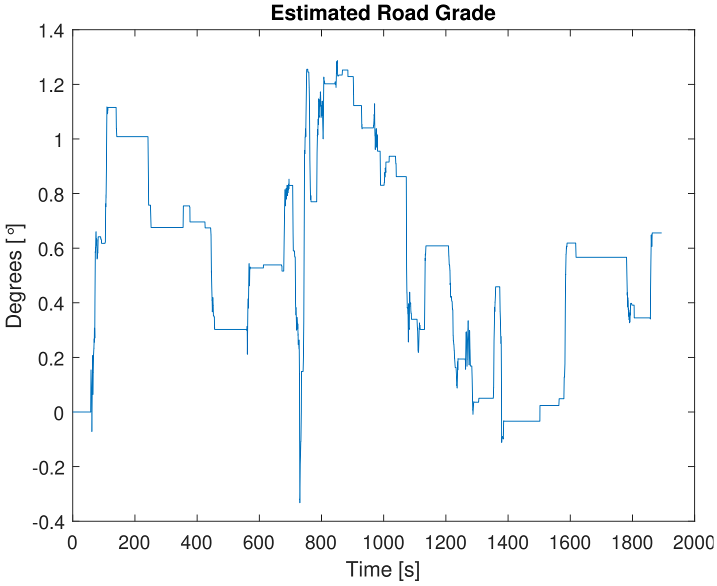


Figure B.32: *Road grade estimate during test 4 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

B.2.4 Test 5

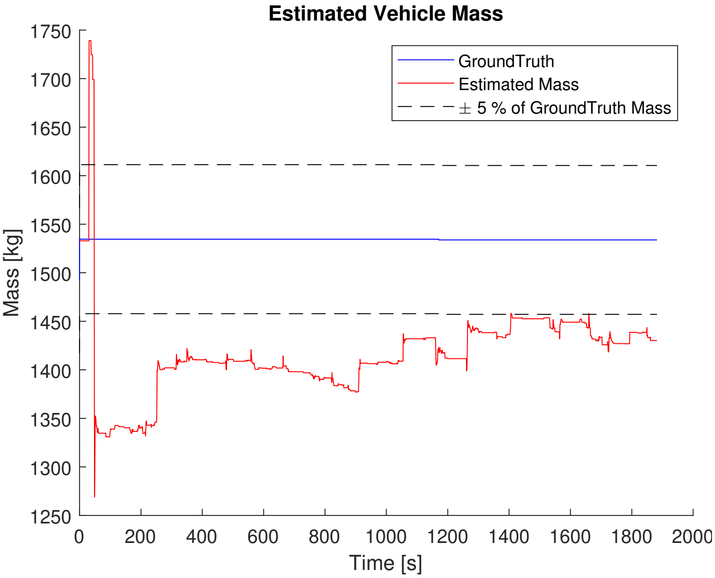


Figure B.33: *Mass estimate during test 5 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

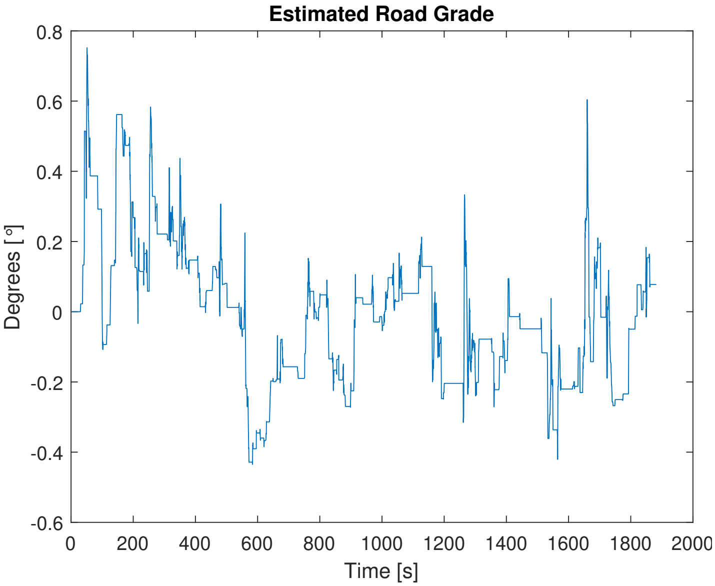


Figure B.34: *Road grade estimate during test 5 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

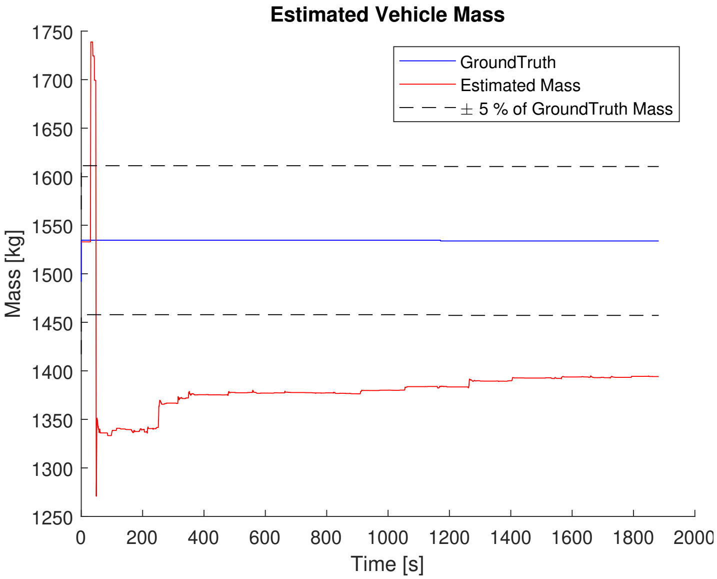


Figure B.35: *Mass estimate during test 5 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

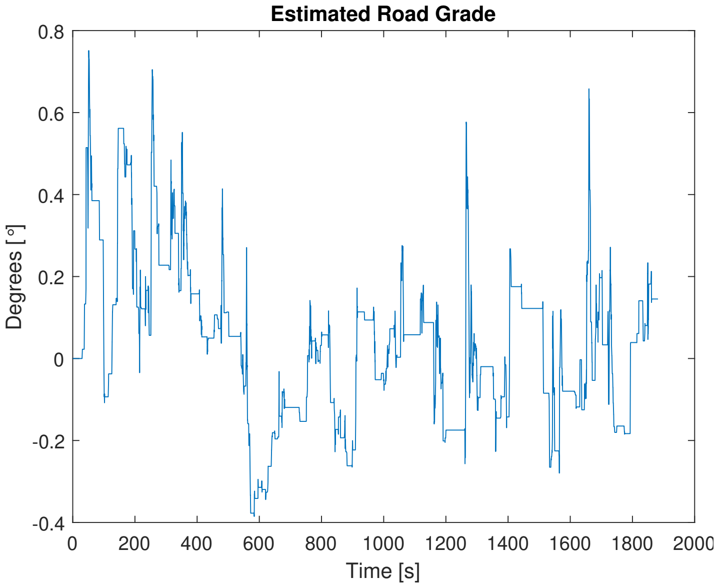


Figure B.36: *Road grade estimate during test 5 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

B.2.5 Test 6

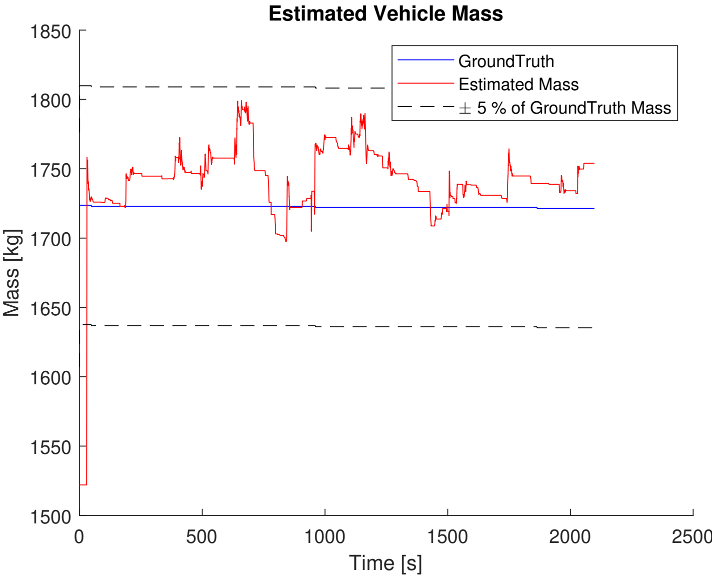


Figure B.37: *Mass estimate during test 6 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

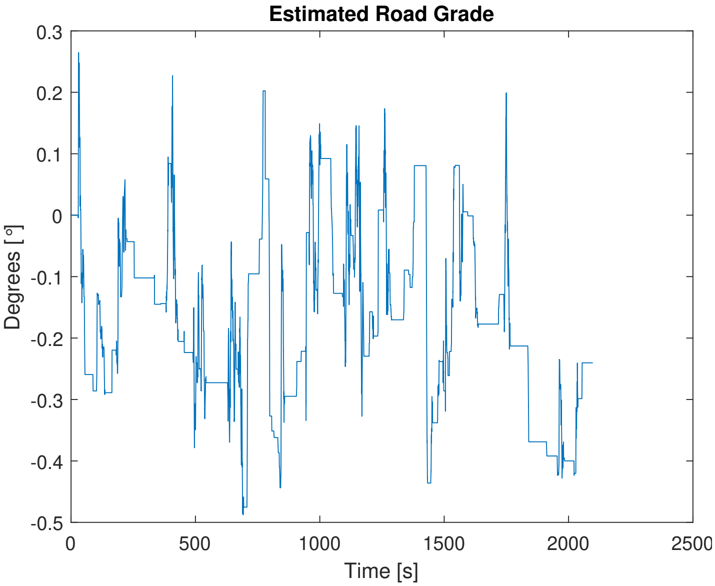


Figure B.38: *Road grade estimate during test 6 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

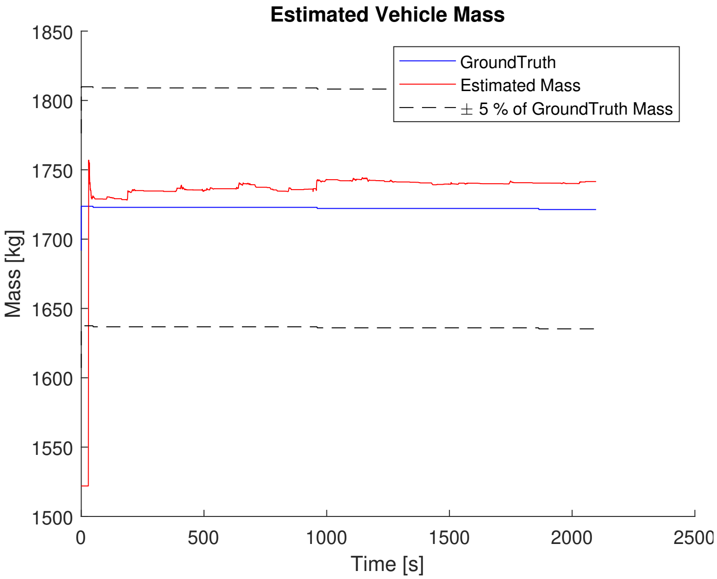


Figure B.39: *Mass estimate during test 6 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

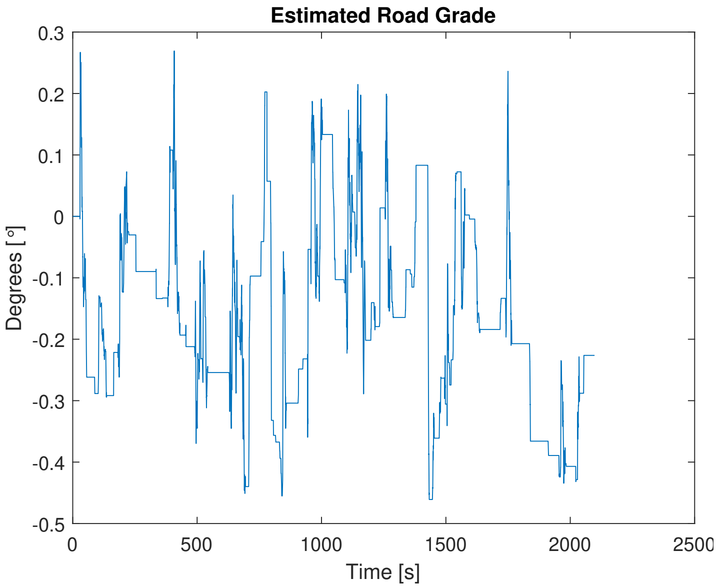


Figure B.40: *Road grade estimate during test 6 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

B.2.6 Test 7

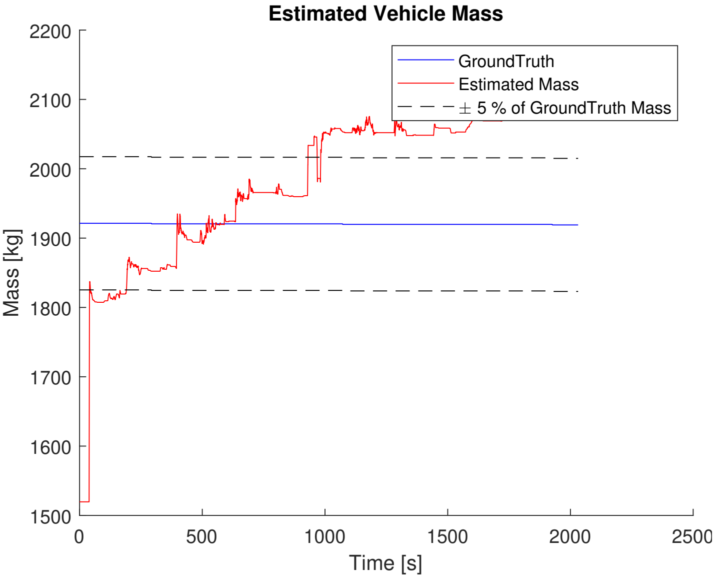


Figure B.41: *Mass estimate during test 7 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

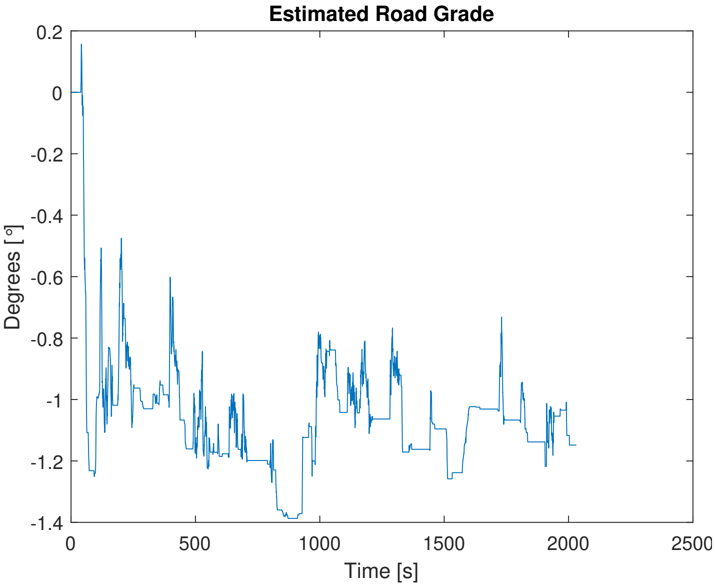


Figure B.42: *Road grade estimate during test 7 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

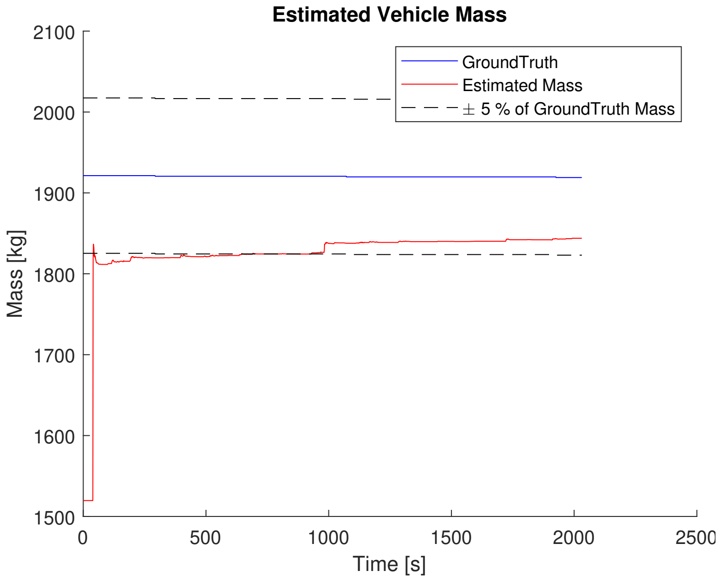


Figure B.43: *Mass estimate during test 7 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

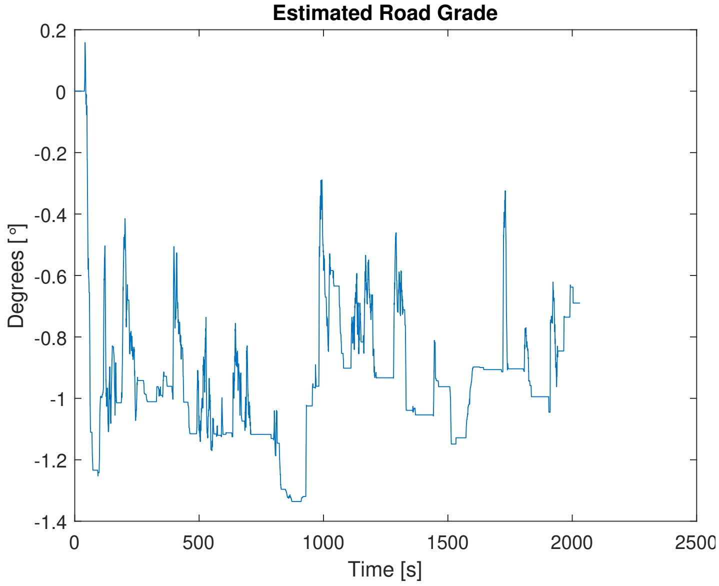


Figure B.44: *Road grade estimate during test 7 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

B.2.7 Test 8

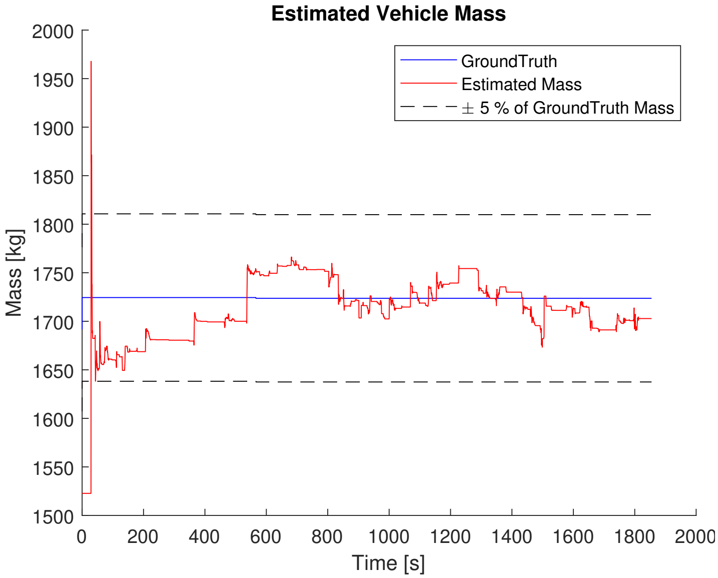


Figure B.45: *Mass estimate during test 8 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

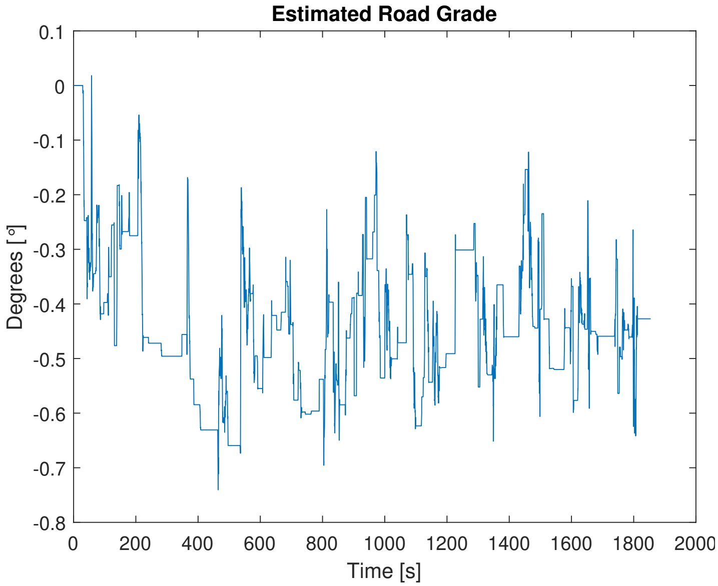


Figure B.46: *Road grade estimate during test 8 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

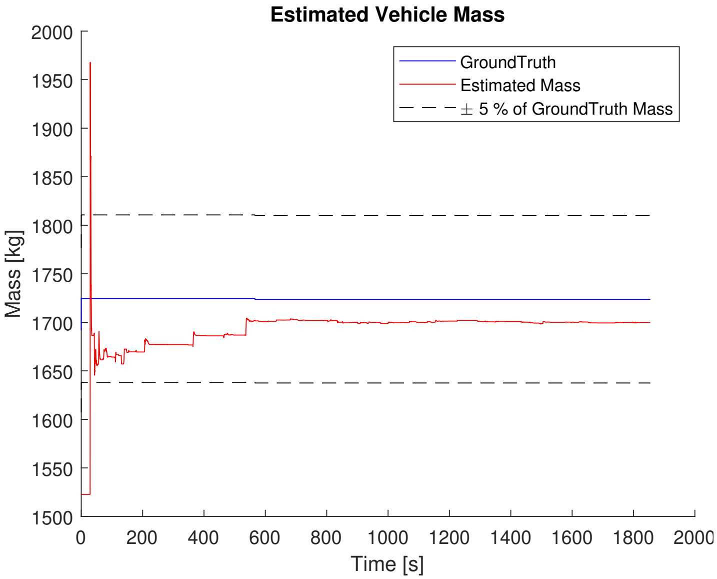


Figure B.47: *Mass estimate during test 8 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

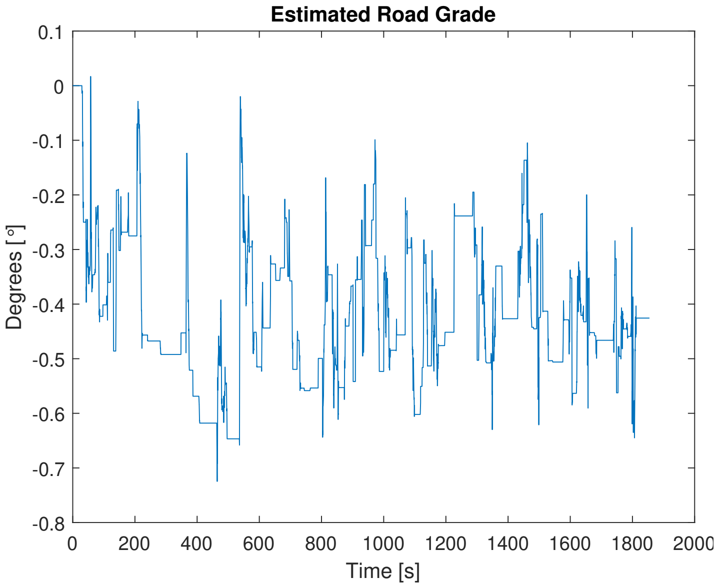


Figure B.48: *Road grade estimate during test 8 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

B.2.8 Test 9

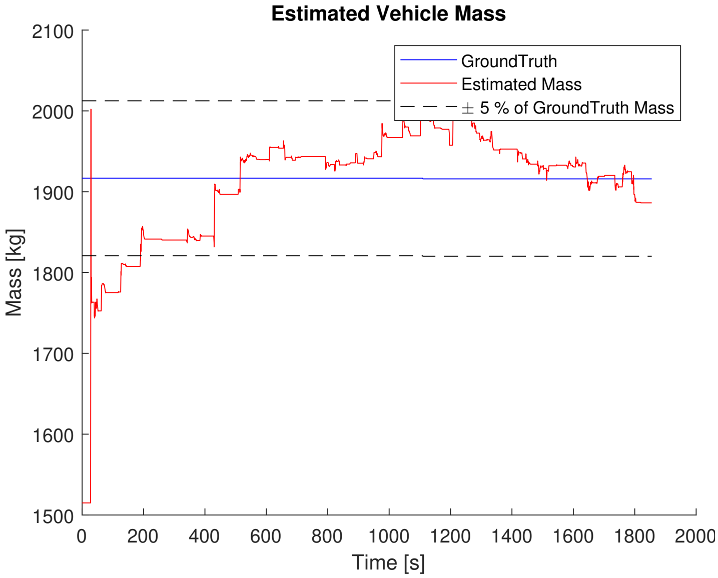


Figure B.49: *Mass estimate during test 9 with multiple forgetting factors* $\lambda_1 = 0.999$*,* $\lambda_2 = 0.99$*.*

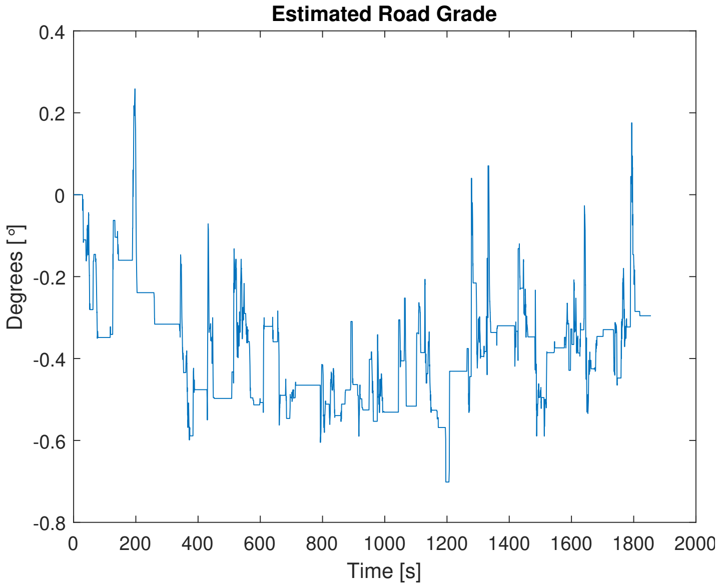


Figure B.50: *Road grade estimate during test 9 with multiple forgetting factors* $\lambda_1 = 0.999$*,* $\lambda_2 = 0.99$*.*

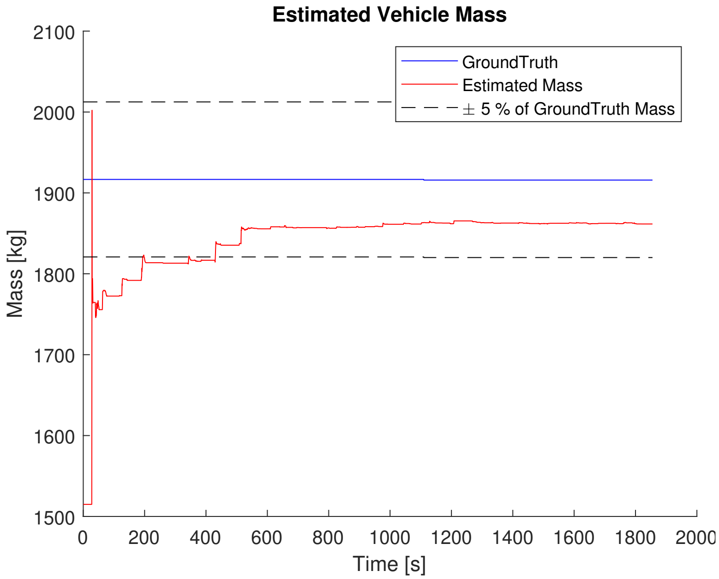


Figure B.51: *Mass estimate during test 9 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

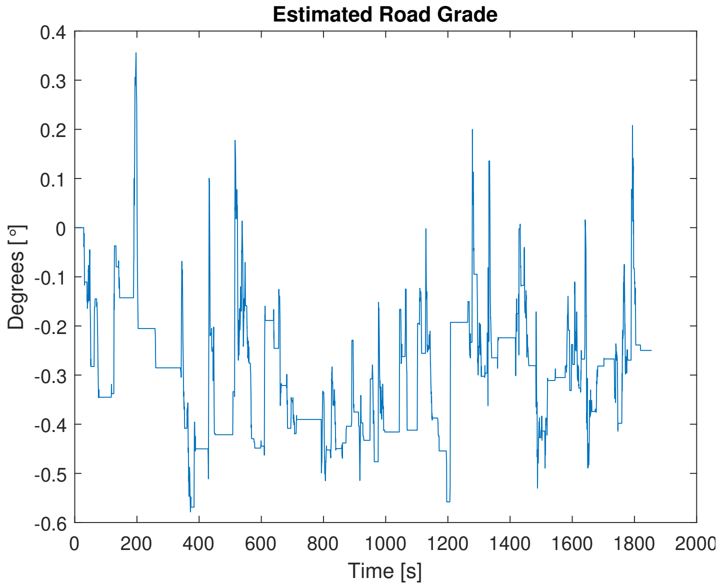


Figure B.52: *Road grade estimate during test 9 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

B.2.9 Test 10

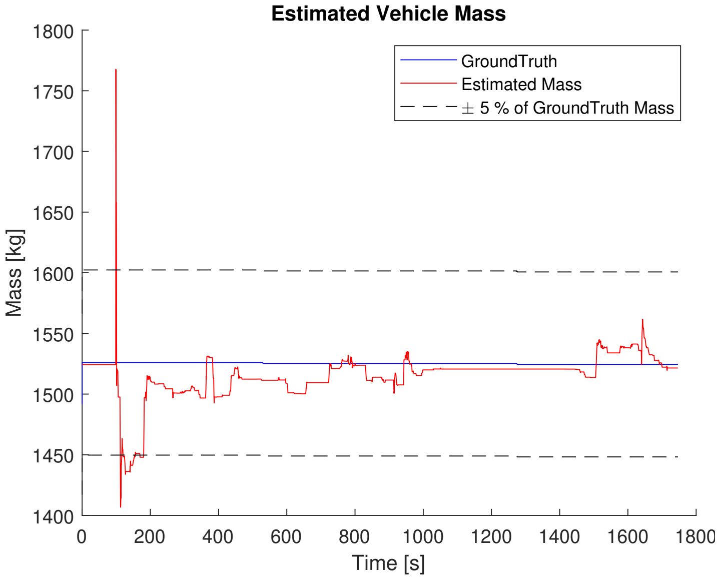


Figure B.53: *Mass estimate during test 10 with multiple forgetting factors* $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.

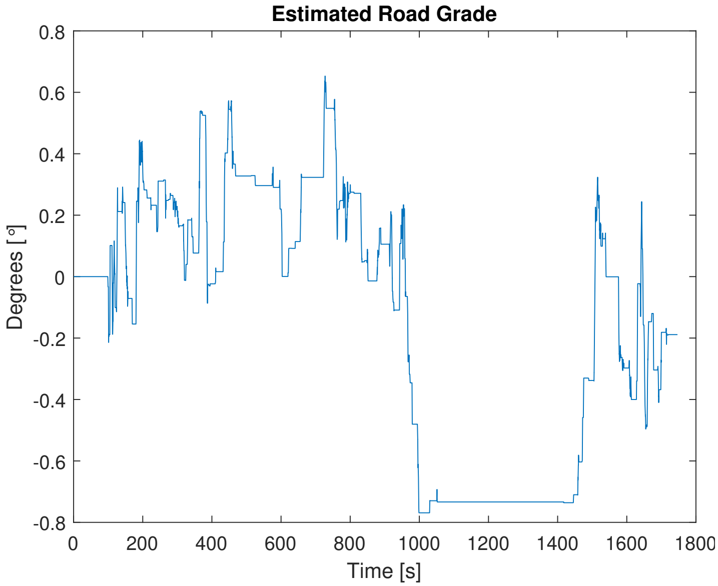


Figure B.54: *Road grade estimate during test 10 with multiple forgetting factors* $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.

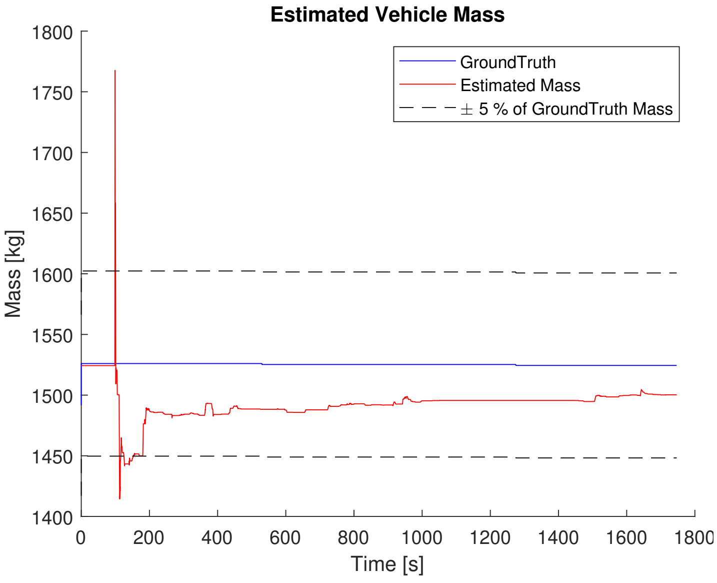


Figure B.55: *Mass estimate during test 10 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

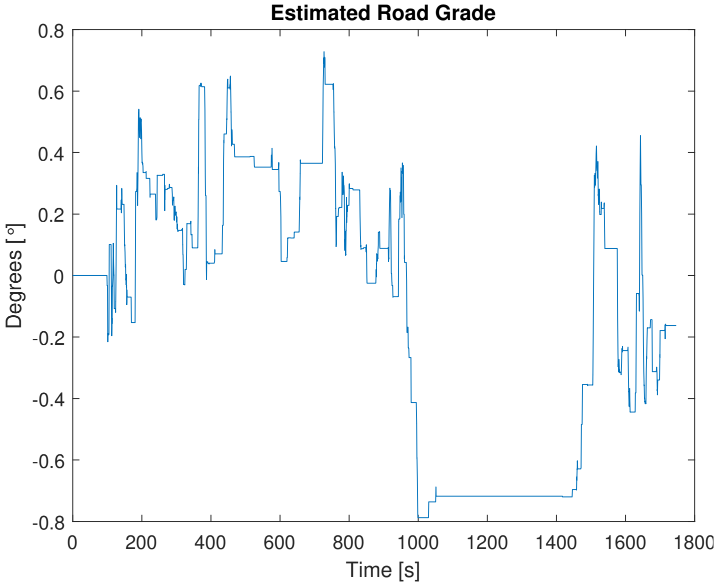


Figure B.56: *Road grade estimate during test 10 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

B.2.10 Test 11

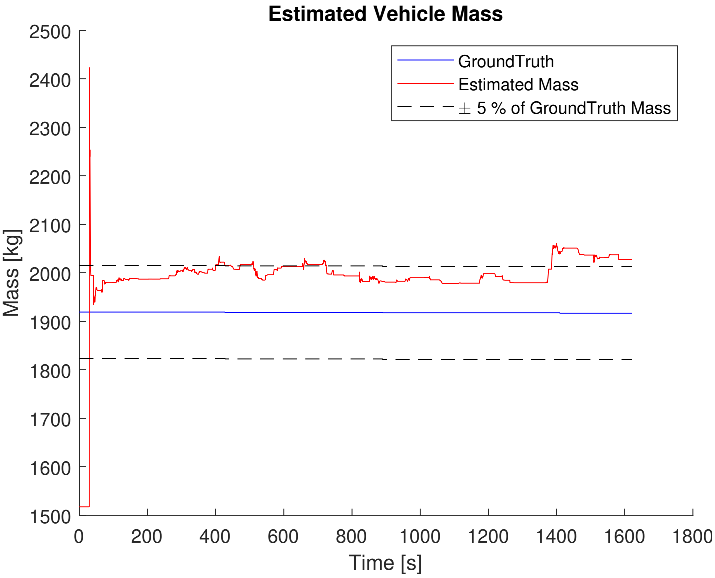


Figure B.57: *Mass estimate during test 11 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

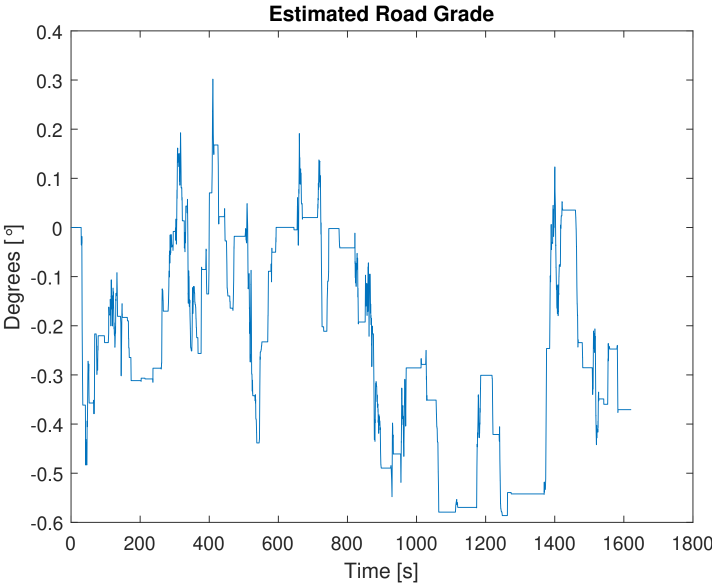


Figure B.58: *Road grade estimate during test 11 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

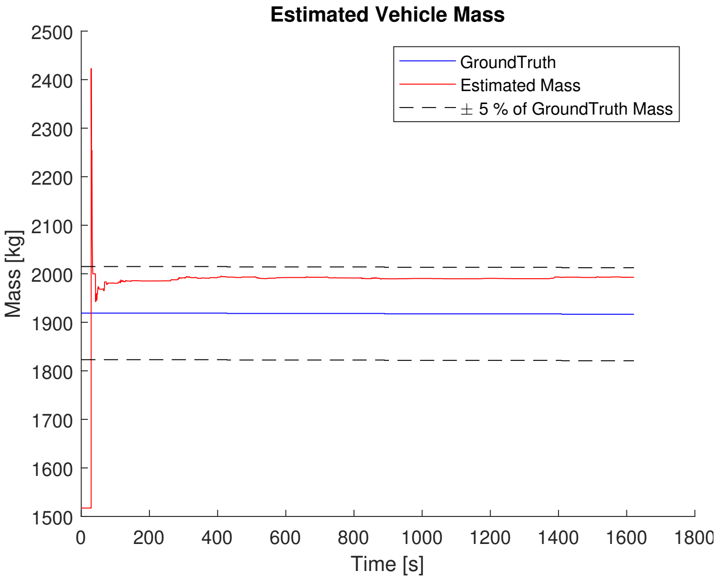


Figure B.59: *Mass estimate during test 11 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

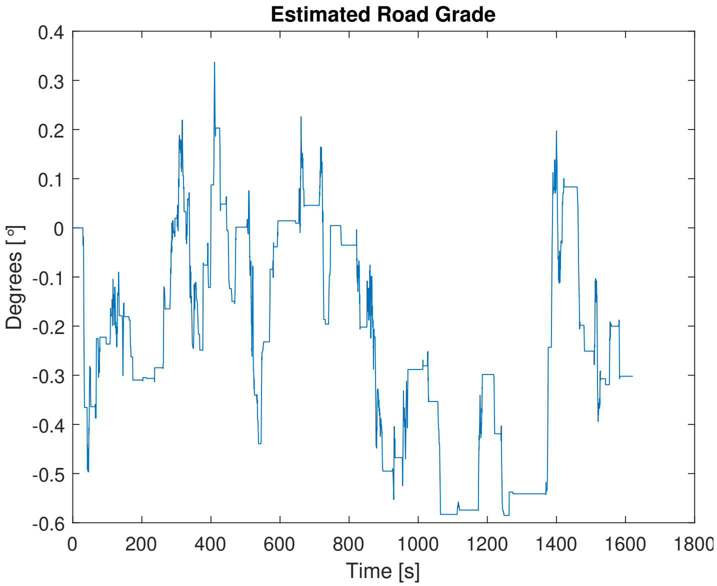


Figure B.60: *Road grade estimate during test 11 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

C
Sensitivity with multiple parameters

This chapter presents plots for the sensitivity with error in wheel radius and rolling resistance.

C.1 Single forgetting factor

This section presents the plots using a single forgetting factor. The forgetting factor is $\lambda = 1$.

C.1.1 Test 2

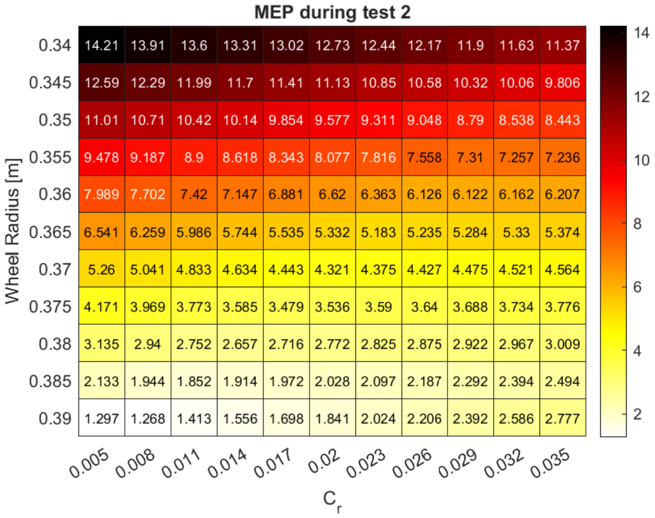


Figure C.1: *Mass estimate during test 2 with a single forgetting factor, $\lambda = 1$.*

C.1.2 Test 3

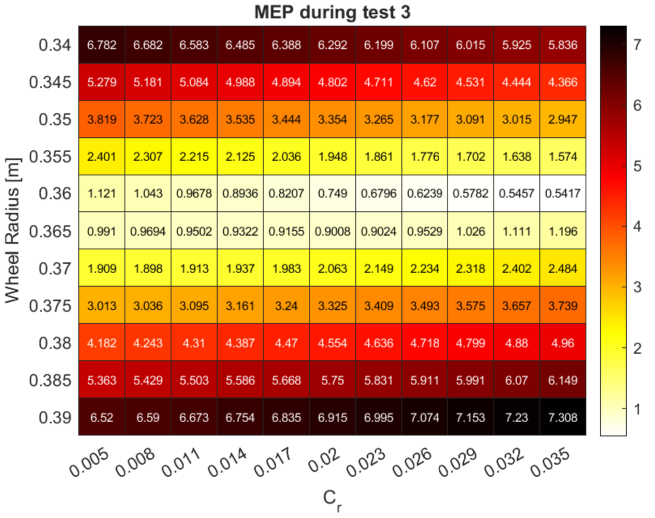
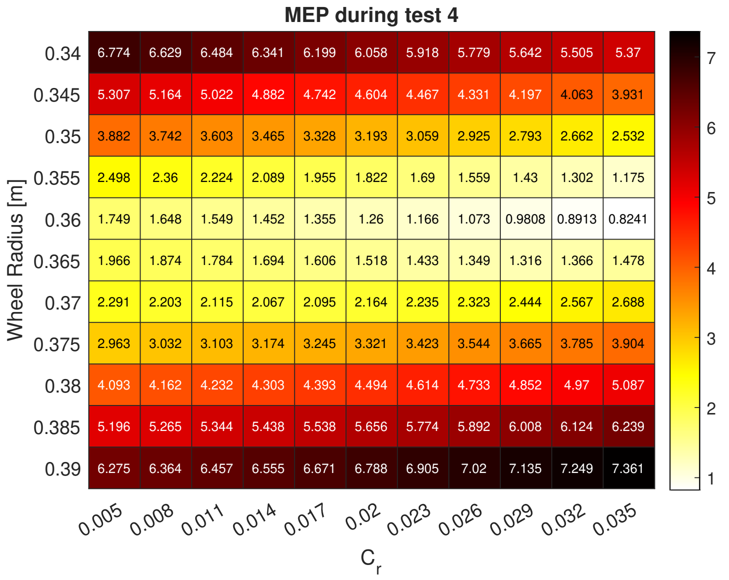


Figure C.2: *Mass estimate during test 3 with a single forgetting factor, $\lambda = 1$.*

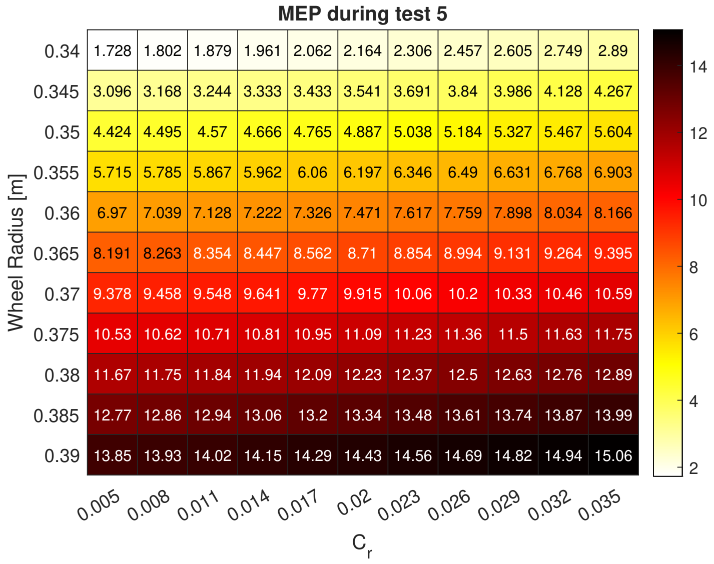
C.1.3 Test 4



Wheel Radius [m] \ C$_r$	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	6.774	6.629	6.484	6.341	6.199	6.058	5.918	5.779	5.642	5.505	5.37
0.345	5.307	5.164	5.022	4.882	4.742	4.604	4.467	4.331	4.197	4.063	3.931
0.35	3.882	3.742	3.603	3.465	3.328	3.193	3.059	2.925	2.793	2.662	2.532
0.355	2.498	2.36	2.224	2.089	1.955	1.822	1.69	1.559	1.43	1.302	1.175
0.36	1.749	1.648	1.549	1.452	1.355	1.26	1.166	1.073	0.9808	0.8913	0.8241
0.365	1.966	1.874	1.784	1.694	1.606	1.518	1.433	1.349	1.316	1.366	1.478
0.37	2.291	2.203	2.115	2.067	2.095	2.164	2.235	2.323	2.444	2.567	2.688
0.375	2.963	3.032	3.103	3.174	3.245	3.321	3.423	3.544	3.665	3.785	3.904
0.38	4.093	4.162	4.232	4.303	4.393	4.494	4.614	4.733	4.852	4.97	5.087
0.385	5.196	5.265	5.344	5.438	5.538	5.656	5.774	5.892	6.008	6.124	6.239
0.39	6.275	6.364	6.457	6.555	6.671	6.788	6.905	7.02	7.135	7.249	7.361

Figure C.3: *Mass estimate during test 4 with a single forgetting factor, $\lambda = 1$.*

C.1.4 Test 5



Wheel Radius [m] \ C$_r$	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	1.728	1.802	1.879	1.961	2.062	2.164	2.306	2.457	2.605	2.749	2.89
0.345	3.096	3.168	3.244	3.333	3.433	3.541	3.691	3.84	3.986	4.128	4.267
0.35	4.424	4.495	4.57	4.666	4.765	4.887	5.038	5.184	5.327	5.467	5.604
0.355	5.715	5.785	5.867	5.962	6.06	6.197	6.346	6.49	6.631	6.768	6.903
0.36	6.97	7.039	7.128	7.222	7.326	7.471	7.617	7.759	7.898	8.034	8.166
0.365	8.191	8.263	8.354	8.447	8.562	8.71	8.854	8.994	9.131	9.264	9.395
0.37	9.378	9.458	9.548	9.641	9.77	9.915	10.06	10.2	10.33	10.46	10.59
0.375	10.53	10.62	10.71	10.81	10.95	11.09	11.23	11.36	11.5	11.63	11.75
0.38	11.67	11.75	11.84	11.94	12.09	12.23	12.37	12.5	12.63	12.76	12.89
0.385	12.77	12.86	12.94	13.06	13.2	13.34	13.48	13.61	13.74	13.87	13.99
0.39	13.85	13.93	14.02	14.15	14.29	14.43	14.56	14.69	14.82	14.94	15.06

Figure C.4: *Mass estimate during test 5 with a single forgetting factor, $\lambda = 1$.*

C.1.5 Test 6

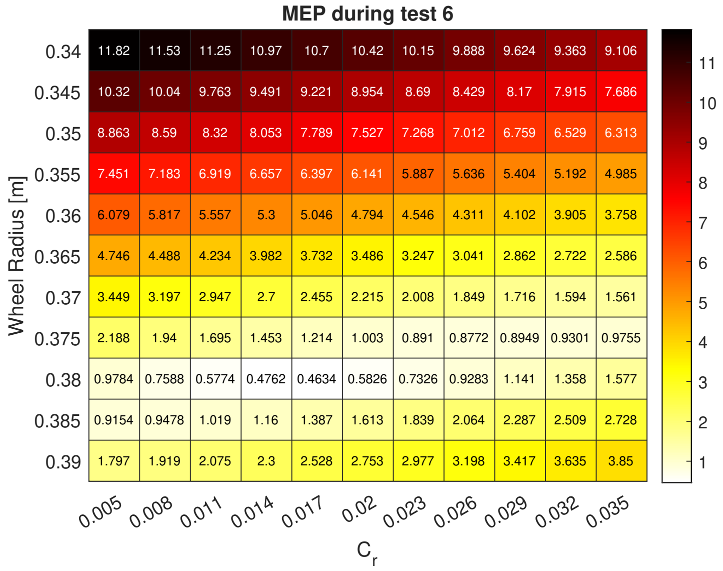


Figure C.5: *Mass estimate during test 6 with a single forgetting factor, $\lambda = 1$.*

C.1.6 Test 7

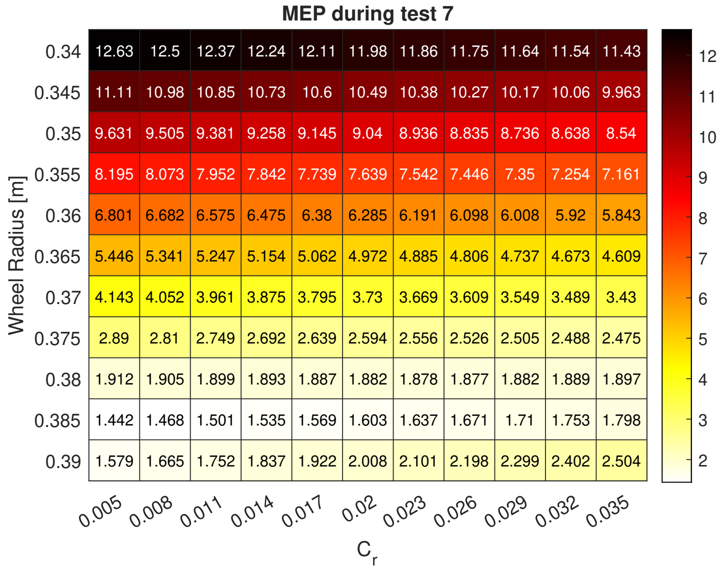
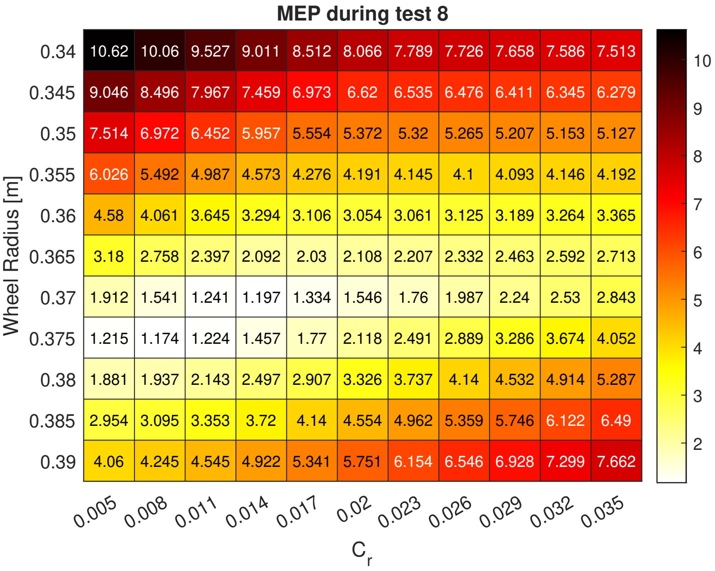


Figure C.6: *Mass estimate during test 7 with a single forgetting factor, $\lambda = 1$.*

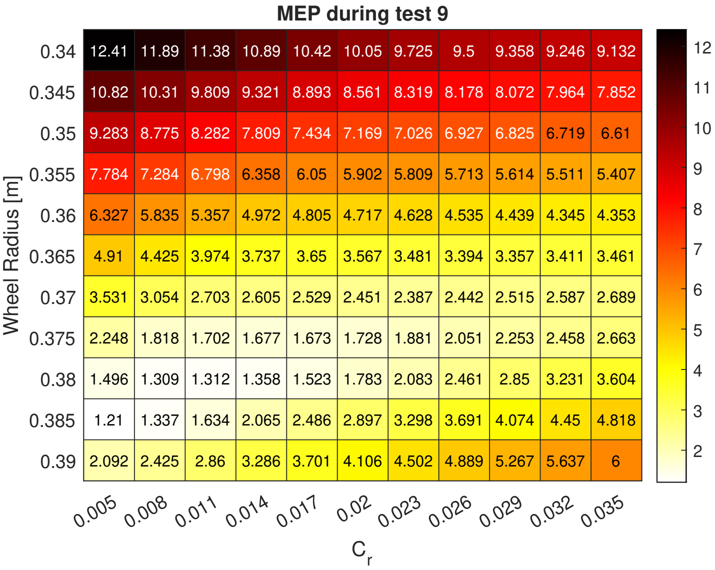
C.1.7 Test 8



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	10.62	10.06	9.527	9.011	8.512	8.066	7.789	7.726	7.658	7.586	7.513
0.345	9.046	8.496	7.967	7.459	6.973	6.62	6.535	6.476	6.411	6.345	6.279
0.35	7.514	6.972	6.452	5.957	5.554	5.372	5.32	5.265	5.207	5.153	5.127
0.355	6.026	5.492	4.987	4.573	4.276	4.191	4.145	4.1	4.093	4.146	4.192
0.36	4.58	4.061	3.645	3.294	3.106	3.054	3.061	3.125	3.189	3.264	3.365
0.365	3.18	2.758	2.397	2.092	2.03	2.108	2.207	2.332	2.463	2.592	2.713
0.37	1.912	1.541	1.241	1.197	1.334	1.546	1.76	1.987	2.24	2.53	2.843
0.375	1.215	1.174	1.224	1.457	1.77	2.118	2.491	2.889	3.286	3.674	4.052
0.38	1.881	1.937	2.143	2.497	2.907	3.326	3.737	4.14	4.532	4.914	5.287
0.385	2.954	3.095	3.353	3.72	4.14	4.554	4.962	5.359	5.746	6.122	6.49
0.39	4.06	4.245	4.545	4.922	5.341	5.751	6.154	6.546	6.928	7.299	7.662

Figure C.7: *Mass estimate during test 8 with a single forgetting factor, $\lambda = 1$.*

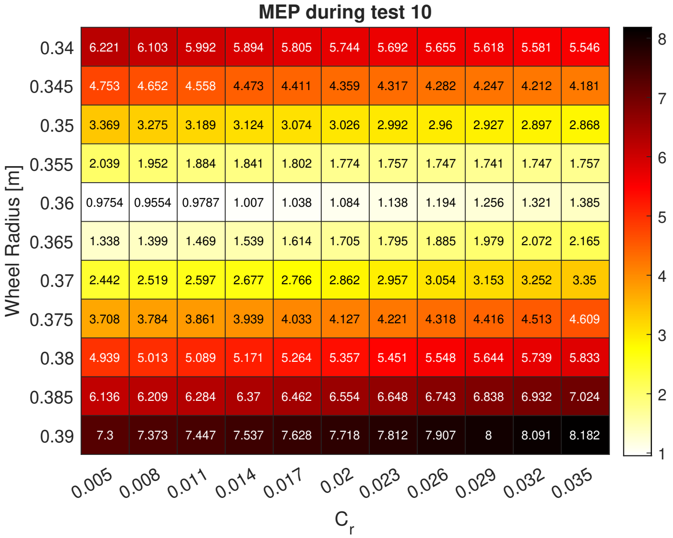
C.1.8 Test 9



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	12.41	11.89	11.38	10.89	10.42	10.05	9.725	9.5	9.358	9.246	9.132
0.345	10.82	10.31	9.809	9.321	8.893	8.561	8.319	8.178	8.072	7.964	7.852
0.35	9.283	8.775	8.282	7.809	7.434	7.169	7.026	6.927	6.825	6.719	6.61
0.355	7.784	7.284	6.798	6.358	6.05	5.902	5.809	5.713	5.614	5.511	5.407
0.36	6.327	5.835	5.357	4.972	4.805	4.717	4.628	4.535	4.439	4.345	4.353
0.365	4.91	4.425	3.974	3.737	3.65	3.567	3.481	3.394	3.357	3.411	3.461
0.37	3.531	3.054	2.703	2.605	2.529	2.451	2.387	2.442	2.515	2.587	2.689
0.375	2.248	1.818	1.702	1.677	1.673	1.728	1.881	2.051	2.253	2.458	2.663
0.38	1.496	1.309	1.312	1.358	1.523	1.783	2.083	2.461	2.85	3.231	3.604
0.385	1.21	1.337	1.634	2.065	2.486	2.897	3.298	3.691	4.074	4.45	4.818
0.39	2.092	2.425	2.86	3.286	3.701	4.106	4.502	4.889	5.267	5.637	6

Figure C.8: *Mass estimate during test 9 with a single forgetting factor, $\lambda = 1$.*

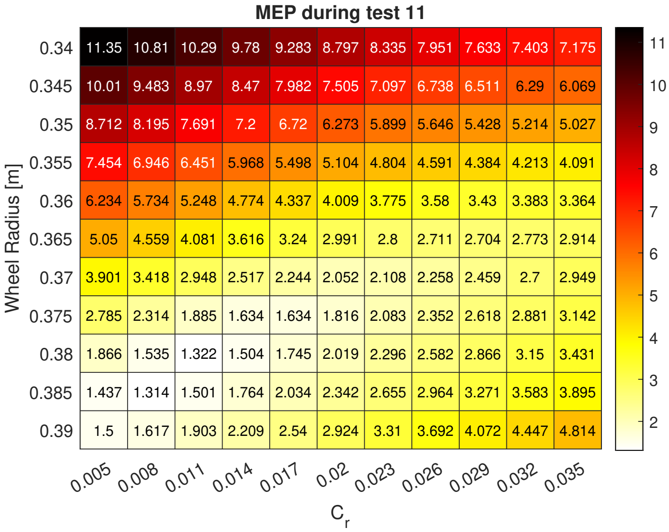
C.1.9 Test 10



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	6.221	6.103	5.992	5.894	5.805	5.744	5.692	5.655	5.618	5.581	5.546
0.345	4.753	4.652	4.558	4.473	4.411	4.359	4.317	4.282	4.247	4.212	4.181
0.35	3.369	3.275	3.189	3.124	3.074	3.026	2.992	2.96	2.927	2.897	2.868
0.355	2.039	1.952	1.884	1.841	1.802	1.774	1.757	1.747	1.741	1.747	1.757
0.36	0.9754	0.9554	0.9787	1.007	1.038	1.084	1.138	1.194	1.256	1.321	1.385
0.365	1.338	1.399	1.469	1.539	1.614	1.705	1.795	1.885	1.979	2.072	2.165
0.37	2.442	2.519	2.597	2.677	2.766	2.862	2.957	3.054	3.153	3.252	3.35
0.375	3.708	3.784	3.861	3.939	4.033	4.127	4.221	4.318	4.416	4.513	4.609
0.38	4.939	5.013	5.089	5.171	5.264	5.357	5.451	5.548	5.644	5.739	5.833
0.385	6.136	6.209	6.284	6.37	6.462	6.554	6.648	6.743	6.838	6.932	7.024
0.39	7.3	7.373	7.447	7.537	7.628	7.718	7.812	7.907	8	8.091	8.182

Figure C.9: *Mass estimate during test 10 with a single forgetting factor, $\lambda = 1$.*

C.1.10 Test 11



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	11.35	10.81	10.29	9.78	9.283	8.797	8.335	7.951	7.633	7.403	7.175
0.345	10.01	9.483	8.97	8.47	7.982	7.505	7.097	6.738	6.511	6.29	6.069
0.35	8.712	8.195	7.691	7.2	6.72	6.273	5.899	5.646	5.428	5.214	5.027
0.355	7.454	6.946	6.451	5.968	5.498	5.104	4.804	4.591	4.384	4.213	4.091
0.36	6.234	5.734	5.248	4.774	4.337	4.009	3.775	3.58	3.43	3.383	3.364
0.365	5.05	4.559	4.081	3.616	3.24	2.991	2.8	2.711	2.704	2.773	2.914
0.37	3.901	3.418	2.948	2.517	2.244	2.052	2.108	2.258	2.459	2.7	2.949
0.375	2.785	2.314	1.885	1.634	1.634	1.816	2.083	2.352	2.618	2.881	3.142
0.38	1.866	1.535	1.322	1.504	1.745	2.019	2.296	2.582	2.866	3.15	3.431
0.385	1.437	1.314	1.501	1.764	2.034	2.342	2.655	2.964	3.271	3.583	3.895
0.39	1.5	1.617	1.903	2.209	2.54	2.924	3.31	3.692	4.072	4.447	4.814

Figure C.10: *Mass estimate during test 11 with a single forgetting factor, $\lambda = 1$.*

C.2 Multiple forgetting factors

The two sets used were $\lambda_1 = 0.999$, $\lambda_2 = 0.99$ and $\lambda_1 = 1$, $\lambda_2 = 0.99$. Both sets are presented under each test.

C.2.1 Test 2

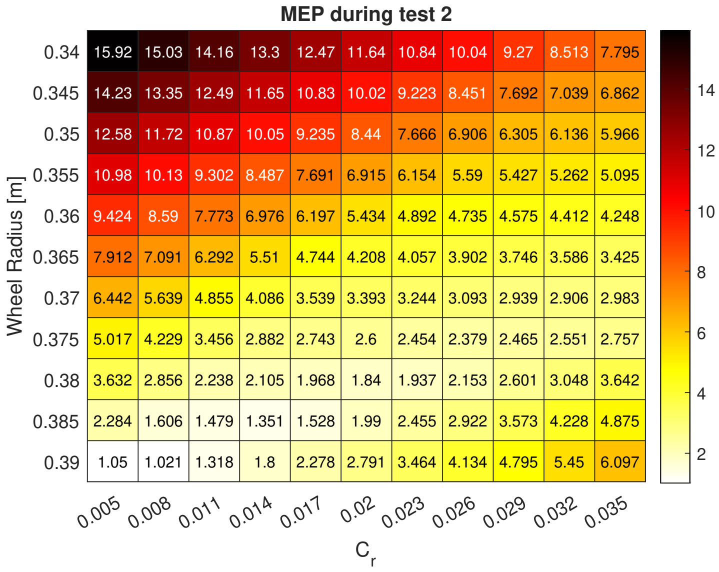


Figure C.11: Sensitivity with multiple parameters for test 2 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.

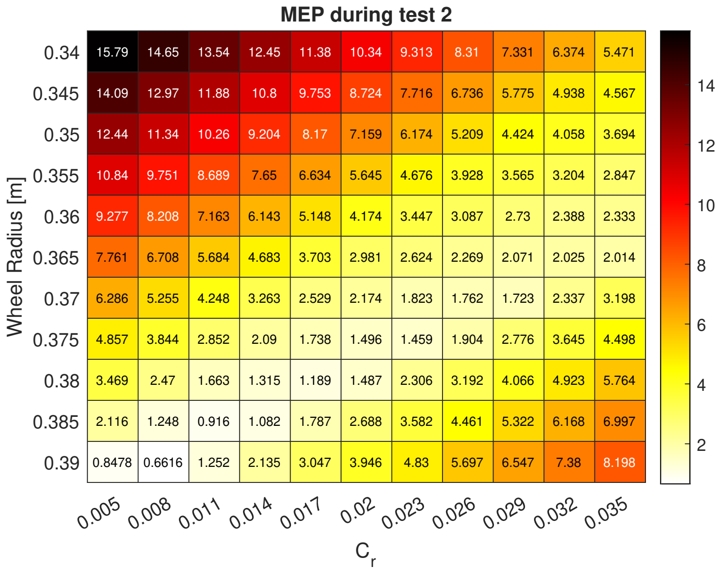
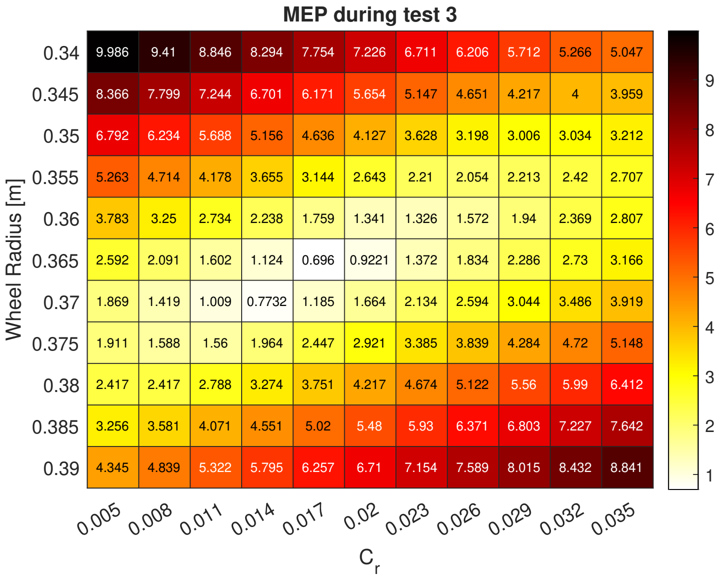


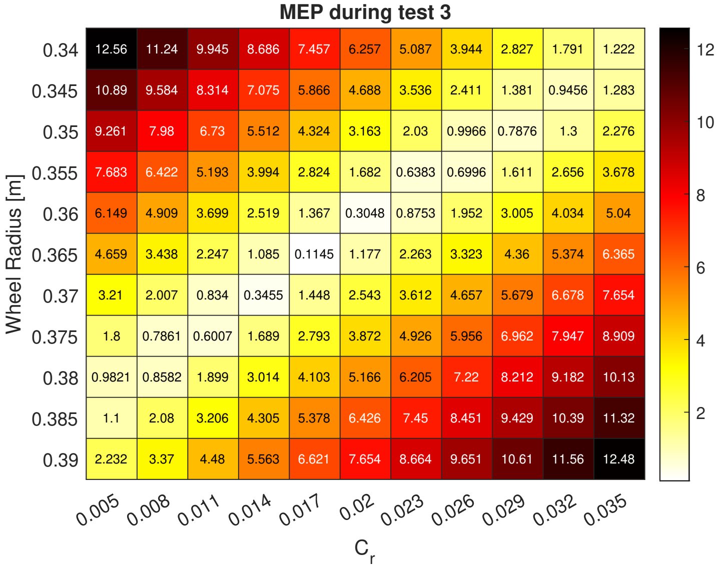
Figure C.12: Sensitivity with multiple parameters for test 2 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.

C.2.2 Test 3



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	9.986	9.41	8.846	8.294	7.754	7.226	6.711	6.206	5.712	5.266	5.047
0.345	8.366	7.799	7.244	6.701	6.171	5.654	5.147	4.651	4.217	4	3.959
0.35	6.792	6.234	5.688	5.156	4.636	4.127	3.628	3.198	3.006	3.034	3.212
0.355	5.263	4.714	4.178	3.655	3.144	2.643	2.21	2.054	2.213	2.42	2.707
0.36	3.783	3.25	2.734	2.238	1.759	1.341	1.326	1.572	1.94	2.369	2.807
0.365	2.592	2.091	1.602	1.124	0.696	0.9221	1.372	1.834	2.286	2.73	3.166
0.37	1.869	1.419	1.009	0.7732	1.185	1.664	2.134	2.594	3.044	3.486	3.919
0.375	1.911	1.588	1.56	1.964	2.447	2.921	3.385	3.839	4.284	4.72	5.148
0.38	2.417	2.417	2.788	3.274	3.751	4.217	4.674	5.122	5.56	5.99	6.412
0.385	3.256	3.581	4.071	4.551	5.02	5.48	5.93	6.371	6.803	7.227	7.642
0.39	4.345	4.839	5.322	5.795	6.257	6.71	7.154	7.589	8.015	8.432	8.841

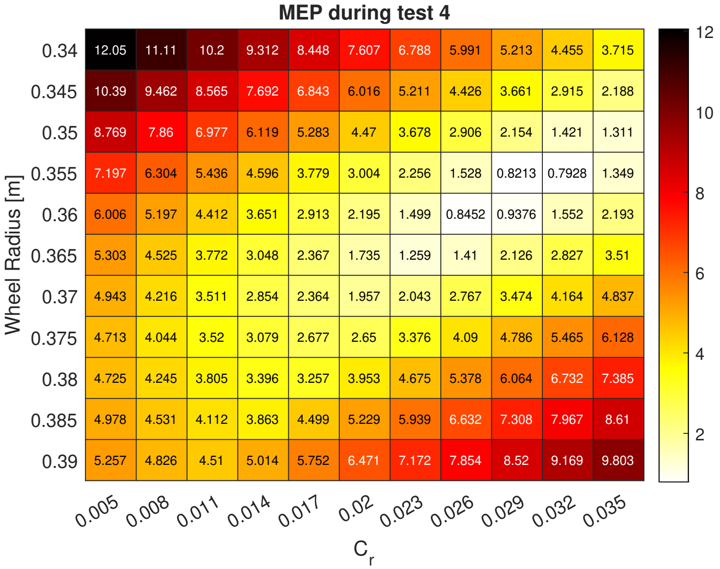
Figure C.13: *Sensitivity with multiple parameters for test 3 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	12.56	11.24	9.945	8.686	7.457	6.257	5.087	3.944	2.827	1.791	1.222
0.345	10.89	9.584	8.314	7.075	5.866	4.688	3.536	2.411	1.381	0.9456	1.283
0.35	9.261	7.98	6.73	5.512	4.324	3.163	2.03	0.9966	0.7876	1.3	2.276
0.355	7.683	6.422	5.193	3.994	2.824	1.682	0.6383	0.6996	1.611	2.656	3.678
0.36	6.149	4.909	3.699	2.519	1.367	0.3048	0.8753	1.952	3.005	4.034	5.04
0.365	4.659	3.438	2.247	1.085	0.1145	1.177	2.263	3.323	4.36	5.374	6.365
0.37	3.21	2.007	0.834	0.3455	1.448	2.543	3.612	4.657	5.679	6.678	7.654
0.375	1.8	0.7861	0.6007	1.689	2.793	3.872	4.926	5.956	6.962	7.947	8.909
0.38	0.9821	0.8582	1.899	3.014	4.103	5.166	6.205	7.22	8.212	9.182	10.13
0.385	1.1	2.08	3.206	4.305	5.378	6.426	7.45	8.451	9.429	10.39	11.32
0.39	2.232	3.37	4.48	5.563	6.621	7.654	8.664	9.651	10.61	11.56	12.48

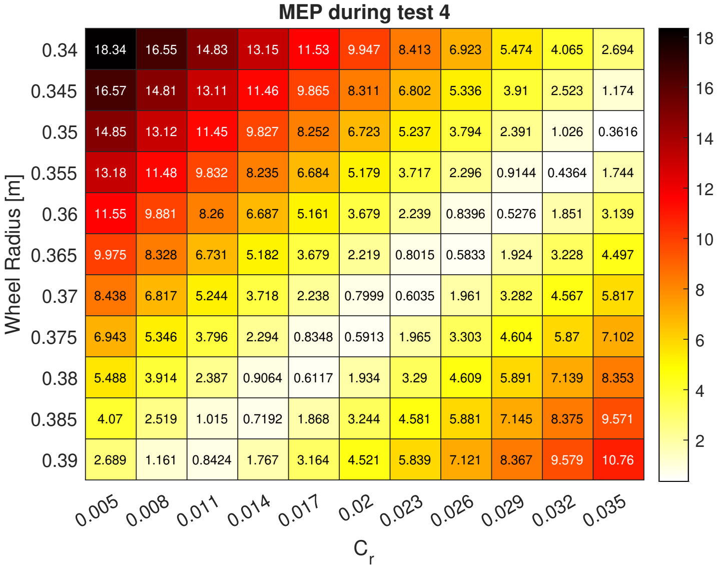
Figure C.14: *Sensitivity with multiple parameters for test 3 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

C.2.3 Test 4



	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	12.05	11.11	10.2	9.312	8.448	7.607	6.788	5.991	5.213	4.455	3.715
0.345	10.39	9.462	8.565	7.692	6.843	6.016	5.211	4.426	3.661	2.915	2.188
0.35	8.769	7.86	6.977	6.119	5.283	4.47	3.678	2.906	2.154	1.421	1.311
0.355	7.197	6.304	5.436	4.596	3.779	3.004	2.256	1.528	0.8213	0.7928	1.349
0.36	6.006	5.197	4.412	3.651	2.913	2.195	1.499	0.8452	0.9376	1.552	2.193
0.365	5.303	4.525	3.772	3.048	2.367	1.735	1.259	1.41	2.126	2.827	3.51
0.37	4.943	4.216	3.511	2.854	2.364	1.957	2.043	2.767	3.474	4.164	4.837
0.375	4.713	4.044	3.52	3.079	2.677	2.65	3.376	4.09	4.786	5.465	6.128
0.38	4.725	4.245	3.805	3.396	3.257	3.953	4.675	5.378	6.064	6.732	7.385
0.385	4.978	4.531	4.112	3.863	4.499	5.229	5.939	6.632	7.308	7.967	8.61
0.39	5.257	4.826	4.51	5.014	5.752	6.471	7.172	7.854	8.52	9.169	9.803

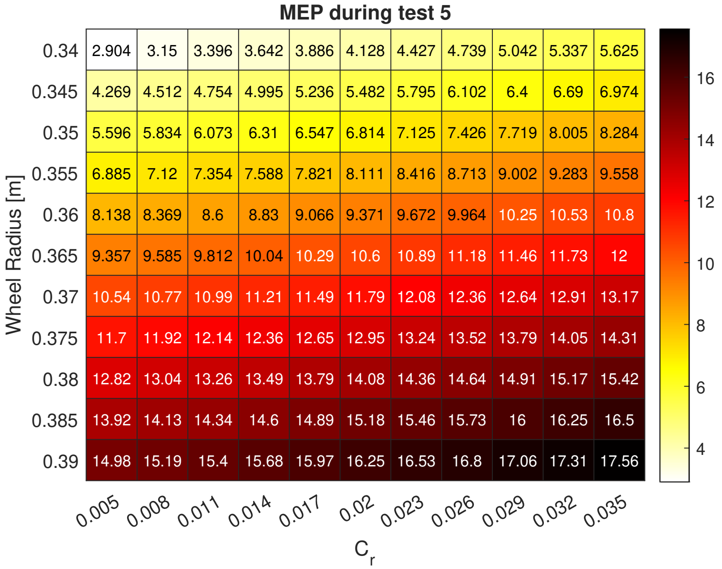
Figure C.15: *Sensitivity with multiple parameters for test 4 with multiple forgetting factors* $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.



	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	18.34	16.55	14.83	13.15	11.53	9.947	8.413	6.923	5.474	4.065	2.694
0.345	16.57	14.81	13.11	11.46	9.865	8.311	6.802	5.336	3.91	2.523	1.174
0.35	14.85	13.12	11.45	9.827	8.252	6.723	5.237	3.794	2.391	1.026	0.3616
0.355	13.18	11.48	9.832	8.235	6.684	5.179	3.717	2.296	0.9144	0.4364	1.744
0.36	11.55	9.881	8.26	6.687	5.161	3.679	2.239	0.8396	0.5276	1.851	3.139
0.365	9.975	8.328	6.731	5.182	3.679	2.219	0.8015	0.5833	1.924	3.228	4.497
0.37	8.438	6.817	5.244	3.718	2.238	0.7999	0.6035	1.961	3.282	4.567	5.817
0.375	6.943	5.346	3.796	2.294	0.8348	0.5913	1.965	3.303	4.604	5.87	7.102
0.38	5.488	3.914	2.387	0.9064	0.6117	1.934	3.29	4.609	5.891	7.139	8.353
0.385	4.07	2.519	1.015	0.7192	1.868	3.244	4.581	5.881	7.145	8.375	9.571
0.39	2.689	1.161	0.8424	1.767	3.164	4.521	5.839	7.121	8.367	9.579	10.76

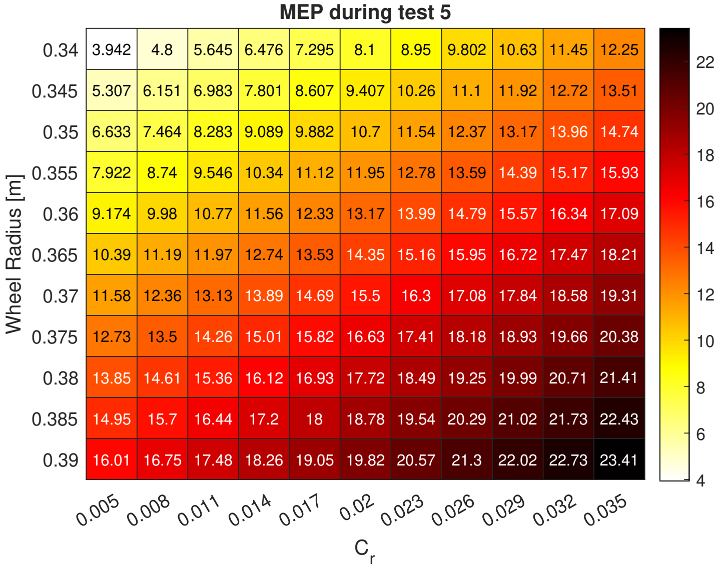
Figure C.16: *Sensitivity with multiple parameters for test 4 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

C.2.4 Test 5



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	2.904	3.15	3.396	3.642	3.886	4.128	4.427	4.739	5.042	5.337	5.625
0.345	4.269	4.512	4.754	4.995	5.236	5.482	5.795	6.102	6.4	6.69	6.974
0.35	5.596	5.834	6.073	6.31	6.547	6.814	7.125	7.426	7.719	8.005	8.284
0.355	6.885	7.12	7.354	7.588	7.821	8.111	8.416	8.713	9.002	9.283	9.558
0.36	8.138	8.369	8.6	8.83	9.066	9.371	9.672	9.964	10.25	10.53	10.8
0.365	9.357	9.585	9.812	10.04	10.29	10.6	10.89	11.18	11.46	11.73	12
0.37	10.54	10.77	10.99	11.21	11.49	11.79	12.08	12.36	12.64	12.91	13.17
0.375	11.7	11.92	12.14	12.36	12.65	12.95	13.24	13.52	13.79	14.05	14.31
0.38	12.82	13.04	13.26	13.49	13.79	14.08	14.36	14.64	14.91	15.17	15.42
0.385	13.92	14.13	14.34	14.6	14.89	15.18	15.46	15.73	16	16.25	16.5
0.39	14.98	15.19	15.4	15.68	15.97	16.25	16.53	16.8	17.06	17.31	17.56

Figure C.17: *Sensitivity with multiple parameters for test 5 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	3.942	4.8	5.645	6.476	7.295	8.1	8.95	9.802	10.63	11.45	12.25
0.345	5.307	6.151	6.983	7.801	8.607	9.407	10.26	11.1	11.92	12.72	13.51
0.35	6.633	7.464	8.283	9.089	9.882	10.7	11.54	12.37	13.17	13.96	14.74
0.355	7.922	8.74	9.546	10.34	11.12	11.95	12.78	13.59	14.39	15.17	15.93
0.36	9.174	9.98	10.77	11.56	12.33	13.17	13.99	14.79	15.57	16.34	17.09
0.365	10.39	11.19	11.97	12.74	13.53	14.35	15.16	15.95	16.72	17.47	18.21
0.37	11.58	12.36	13.13	13.89	14.69	15.5	16.3	17.08	17.84	18.58	19.31
0.375	12.73	13.5	14.26	15.01	15.82	16.63	17.41	18.18	18.93	19.66	20.38
0.38	13.85	14.61	15.36	16.12	16.93	17.72	18.49	19.25	19.99	20.71	21.41
0.385	14.95	15.7	16.44	17.2	18	18.78	19.54	20.29	21.02	21.73	22.43
0.39	16.01	16.75	17.48	18.26	19.05	19.82	20.57	21.3	22.02	22.73	23.41

Figure C.18: *Sensitivity with multiple parameters for test 5 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

C.2.5 Test 6

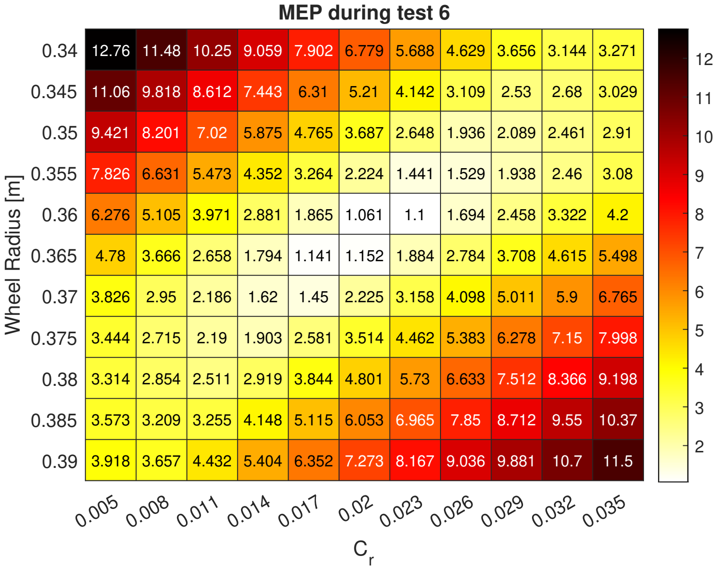


Figure C.19: *Sensitivity with multiple parameters for test 6 with multiple forgetting factors* $\lambda_1 = 0.999$*,* $\lambda_2 = 0.99$*.*

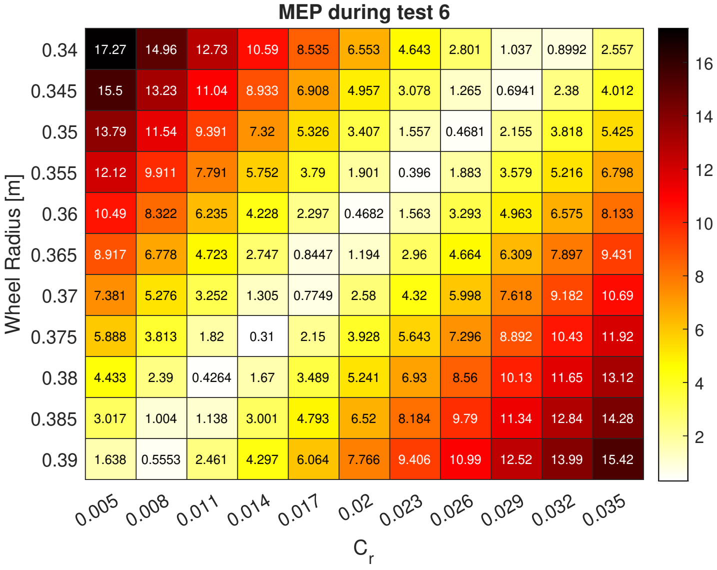


Figure C.20: *Sensitivity with multiple parameters for test 6 with multiple forgetting factors* $\lambda_1 = 1$*,* $\lambda_2 = 0.99$*.*

C.2.6 Test 7

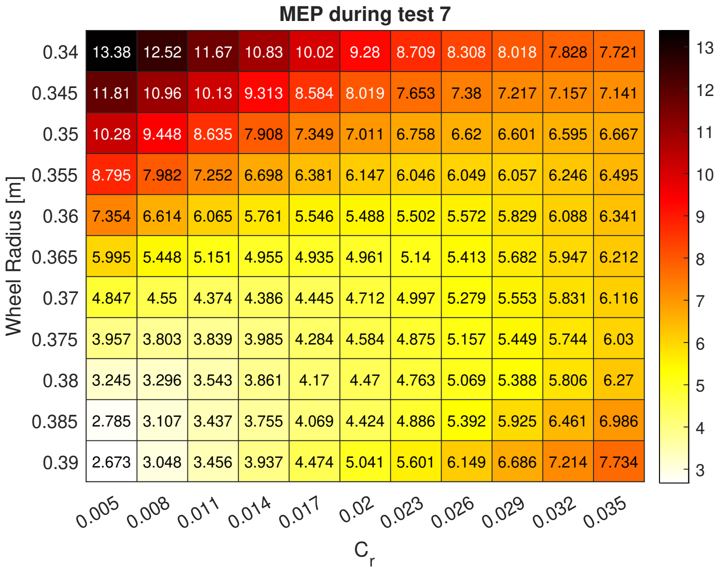


Figure C.21: *Sensitivity with multiple parameters for test 7 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

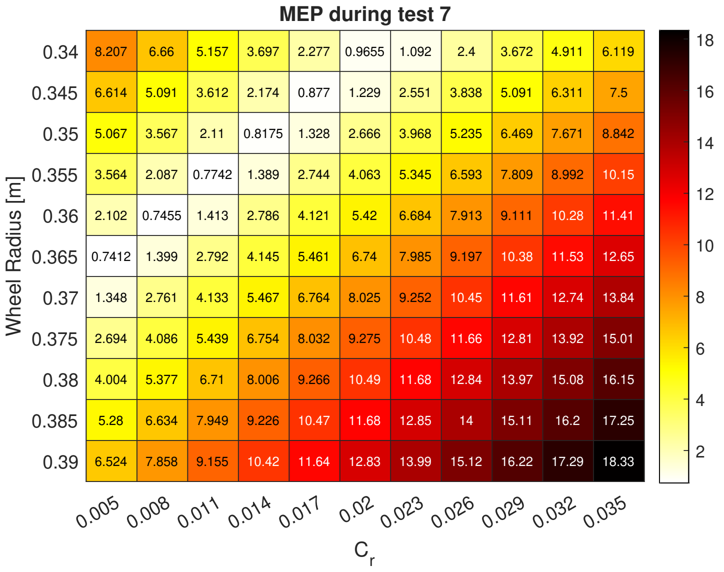
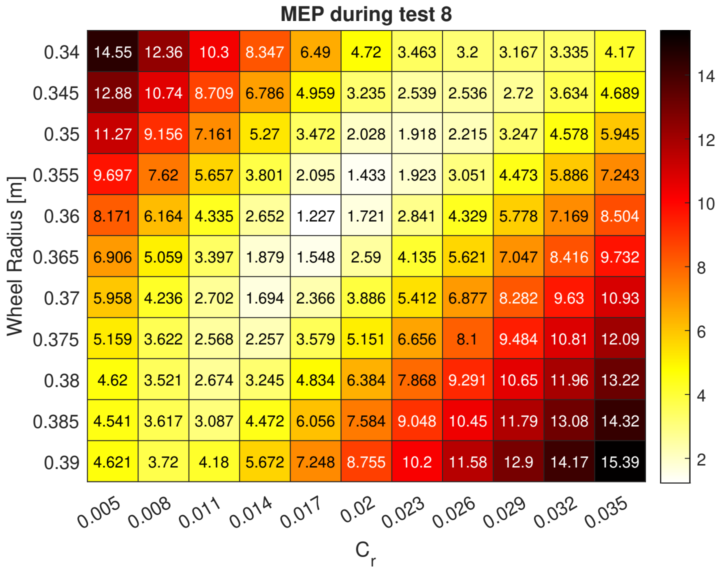


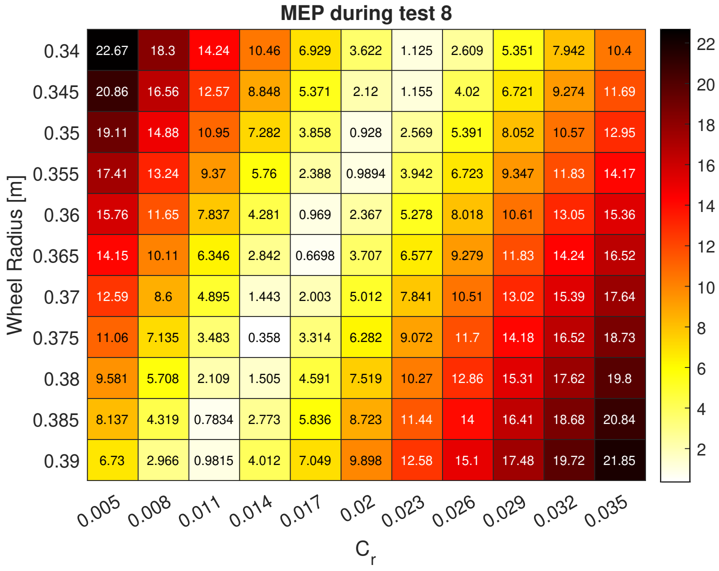
Figure C.22: *Sensitivity with multiple parameters for test 7 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

C.2.7 Test 8



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	14.55	12.36	10.3	8.347	6.49	4.72	3.463	3.2	3.167	3.335	4.17
0.345	12.88	10.74	8.709	6.786	4.959	3.235	2.539	2.536	2.72	3.634	4.689
0.35	11.27	9.156	7.161	5.27	3.472	2.028	1.918	2.215	3.247	4.578	5.945
0.355	9.697	7.62	5.657	3.801	2.095	1.433	1.923	3.051	4.473	5.886	7.243
0.36	8.171	6.164	4.335	2.652	1.227	1.721	2.841	4.329	5.778	7.169	8.504
0.365	6.906	5.059	3.397	1.879	1.548	2.59	4.135	5.621	7.047	8.416	9.732
0.37	5.958	4.236	2.702	1.694	2.366	3.886	5.412	6.877	8.282	9.63	10.93
0.375	5.159	3.622	2.568	2.257	3.579	5.151	6.656	8.1	9.484	10.81	12.09
0.38	4.62	3.521	2.674	3.245	4.834	6.384	7.868	9.291	10.65	11.96	13.22
0.385	4.541	3.617	3.087	4.472	6.056	7.584	9.048	10.45	11.79	13.08	14.32
0.39	4.621	3.72	4.18	5.672	7.248	8.755	10.2	11.58	12.9	14.17	15.39

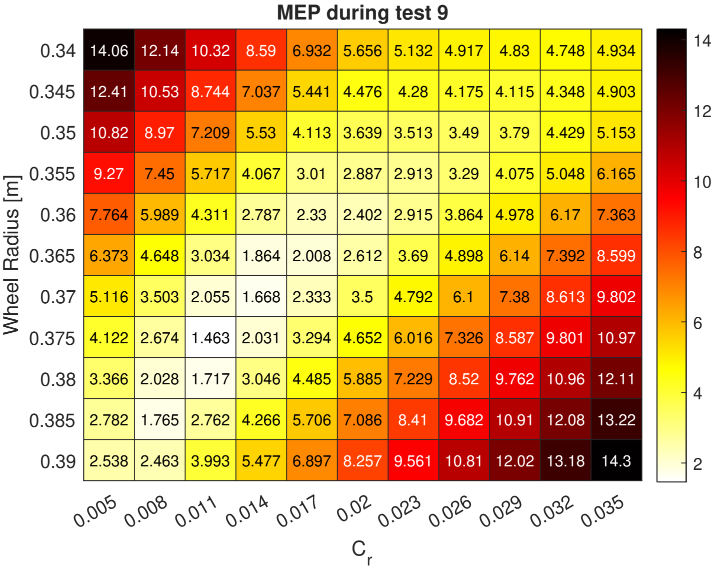
Figure C.23: *Sensitivity with multiple parameters for test 8 with multiple forgetting factors* $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.



Wheel Radius [m] \ C_r	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	22.67	18.3	14.24	10.46	6.929	3.622	1.125	2.609	5.351	7.942	10.4
0.345	20.86	16.56	12.57	8.848	5.371	2.12	1.155	4.02	6.721	9.274	11.69
0.35	19.11	14.88	10.95	7.282	3.858	0.928	2.569	5.391	8.052	10.57	12.95
0.355	17.41	13.24	9.37	5.76	2.388	0.9894	3.942	6.723	9.347	11.83	14.17
0.36	15.76	11.65	7.837	4.281	0.969	2.367	5.278	8.018	10.61	13.05	15.36
0.365	14.15	10.11	6.346	2.842	0.6698	3.707	6.577	9.279	11.83	14.24	16.52
0.37	12.59	8.6	4.895	1.443	2.003	5.012	7.841	10.51	13.02	15.39	17.64
0.375	11.06	7.135	3.483	0.358	3.314	6.282	9.072	11.7	14.18	16.52	18.73
0.38	9.581	5.708	2.109	1.505	4.591	7.519	10.27	12.86	15.31	17.62	19.8
0.385	8.137	4.319	0.7834	2.773	5.836	8.723	11.44	14	16.41	18.68	20.84
0.39	6.73	2.966	0.9815	4.012	7.049	9.898	12.58	15.1	17.48	19.72	21.85

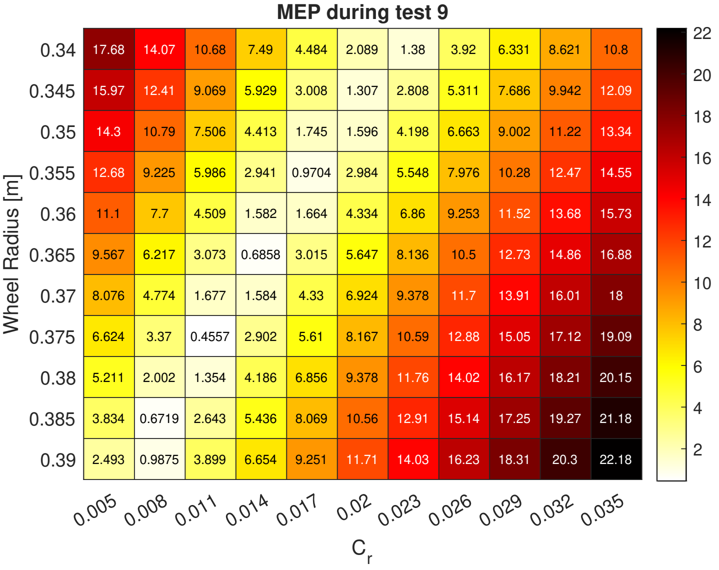
Figure C.24: *Sensitivity with multiple parameters for test 8 with multiple forgetting factors* $\lambda_1 = 1$, $\lambda_2 = 0.99$.

C.2.8 Test 9



	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	14.06	12.14	10.32	8.59	6.932	5.656	5.132	4.917	4.83	4.748	4.934
0.345	12.41	10.53	8.744	7.037	5.441	4.476	4.28	4.175	4.115	4.348	4.903
0.35	10.82	8.97	7.209	5.53	4.113	3.639	3.513	3.49	3.79	4.429	5.153
0.355	9.27	7.45	5.717	4.067	3.01	2.887	2.913	3.29	4.075	5.048	6.165
0.36	7.764	5.989	4.311	2.787	2.33	2.402	2.915	3.864	4.978	6.17	7.363
0.365	6.373	4.648	3.034	1.864	2.008	2.612	3.69	4.898	6.14	7.392	8.599
0.37	5.116	3.503	2.055	1.668	2.333	3.5	4.792	6.1	7.38	8.613	9.802
0.375	4.122	2.674	1.463	2.031	3.294	4.652	6.016	7.326	8.587	9.801	10.97
0.38	3.366	2.028	1.717	3.046	4.485	5.885	7.229	8.52	9.762	10.96	12.11
0.385	2.782	1.765	2.762	4.266	5.706	7.086	8.41	9.682	10.91	12.08	13.22
0.39	2.538	2.463	3.993	5.477	6.897	8.257	9.561	10.81	12.02	13.18	14.3

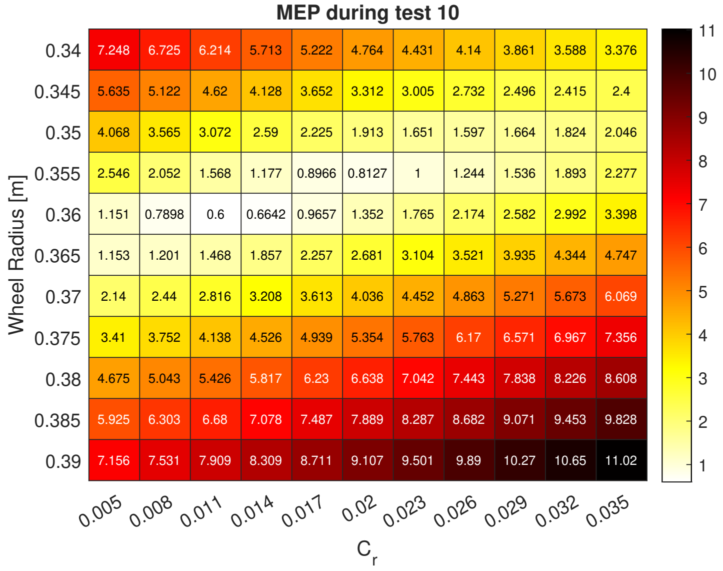
Figure C.25: *Sensitivity with multiple parameters for test 9 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*



	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	17.68	14.07	10.68	7.49	4.484	2.089	1.38	3.92	6.331	8.621	10.8
0.345	15.97	12.41	9.069	5.929	3.008	1.307	2.808	5.311	7.686	9.942	12.09
0.35	14.3	10.79	7.506	4.413	1.745	1.596	4.198	6.663	9.002	11.22	13.34
0.355	12.68	9.225	5.986	2.941	0.9704	2.984	5.548	7.976	10.28	12.47	14.55
0.36	11.1	7.7	4.509	1.582	1.664	4.334	6.86	9.253	11.52	13.68	15.73
0.365	9.567	6.217	3.073	0.6858	3.015	5.647	8.136	10.5	12.73	14.86	16.88
0.37	8.076	4.774	1.677	1.584	4.33	6.924	9.378	11.7	13.91	16.01	18
0.375	6.624	3.37	0.4557	2.902	5.61	8.167	10.59	12.88	15.05	17.12	19.09
0.38	5.211	2.002	1.354	4.186	6.856	9.378	11.76	14.02	16.17	18.21	20.15
0.385	3.834	0.6719	2.643	5.436	8.069	10.56	12.91	15.14	17.25	19.27	21.18
0.39	2.493	0.9875	3.899	6.654	9.251	11.71	14.03	16.23	18.31	20.3	22.18

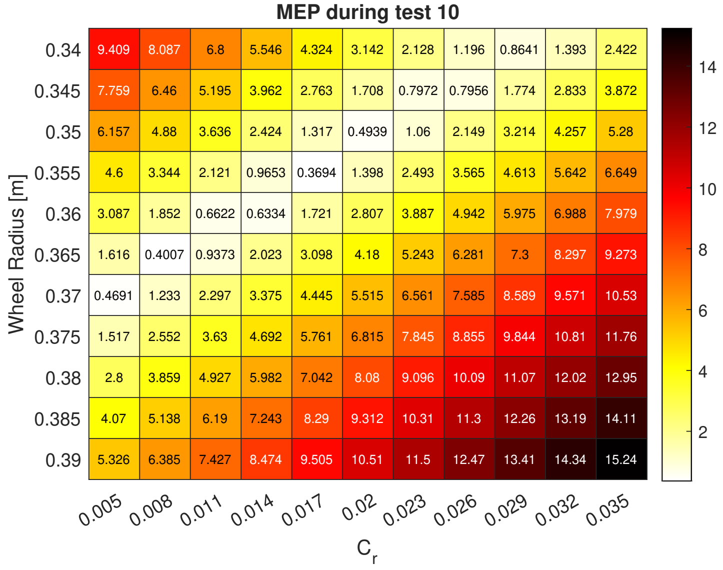
Figure C.26: *Sensitivity with multiple parameters for test 9 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

C.2.9 Test 10



Wheel Radius [m] \ C$_r$	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	7.248	6.725	6.214	5.713	5.222	4.764	4.431	4.14	3.861	3.588	3.376
0.345	5.635	5.122	4.62	4.128	3.652	3.312	3.005	2.732	2.496	2.415	2.4
0.35	4.068	3.565	3.072	2.59	2.225	1.913	1.651	1.597	1.664	1.824	2.046
0.355	2.546	2.052	1.568	1.177	0.8966	0.8127	1	1.244	1.536	1.893	2.277
0.36	1.151	0.7898	0.6	0.6642	0.9657	1.352	1.765	2.174	2.582	2.992	3.398
0.365	1.153	1.201	1.468	1.857	2.257	2.681	3.104	3.521	3.935	4.344	4.747
0.37	2.14	2.44	2.816	3.208	3.613	4.036	4.452	4.863	5.271	5.673	6.069
0.375	3.41	3.752	4.138	4.526	4.939	5.354	5.763	6.17	6.571	6.967	7.356
0.38	4.675	5.043	5.426	5.817	6.23	6.638	7.042	7.443	7.838	8.226	8.608
0.385	5.925	6.303	6.68	7.078	7.487	7.889	8.287	8.682	9.071	9.453	9.828
0.39	7.156	7.531	7.909	8.309	8.711	9.107	9.501	9.89	10.27	10.65	11.02

Figure C.27: *Sensitivity with multiple parameters for test 10 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*



Wheel Radius [m] \ C$_r$	0.005	0.008	0.011	0.014	0.017	0.02	0.023	0.026	0.029	0.032	0.035
0.34	9.409	8.087	6.8	5.546	4.324	3.142	2.128	1.196	0.8641	1.393	2.422
0.345	7.759	6.46	5.195	3.962	2.763	1.708	0.7972	0.7956	1.774	2.833	3.872
0.35	6.157	4.88	3.636	2.424	1.317	0.4939	1.06	2.149	3.214	4.257	5.28
0.355	4.6	3.344	2.121	0.9653	0.3694	1.398	2.493	3.565	4.613	5.642	6.649
0.36	3.087	1.852	0.6622	0.6334	1.721	2.807	3.887	4.942	5.975	6.988	7.979
0.365	1.616	0.4007	0.9373	2.023	3.098	4.18	5.243	6.281	7.3	8.297	9.273
0.37	0.4691	1.233	2.297	3.375	4.445	5.515	6.561	7.585	8.589	9.571	10.53
0.375	1.517	2.552	3.63	4.692	5.761	6.815	7.845	8.855	9.844	10.81	11.76
0.38	2.8	3.859	4.927	5.982	7.042	8.08	9.096	10.09	11.07	12.02	12.95
0.385	4.07	5.138	6.19	7.243	8.29	9.312	10.31	11.3	12.26	13.19	14.11
0.39	5.326	6.385	7.427	8.474	9.505	10.51	11.5	12.47	13.41	14.34	15.24

Figure C.28: *Sensitivity with multiple parameters for test 10 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

C.2.10 Test 11

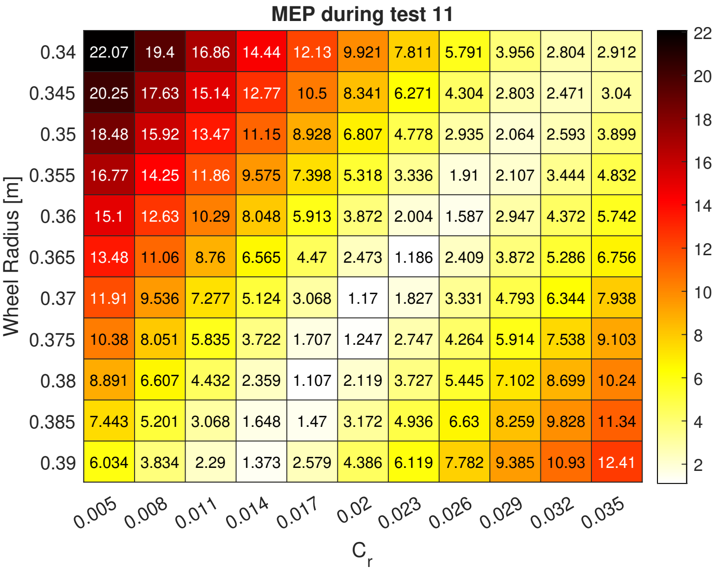


Figure C.29: *Sensitivity with multiple parameters for test 11 with multiple forgetting factors $\lambda_1 = 0.999$, $\lambda_2 = 0.99$.*

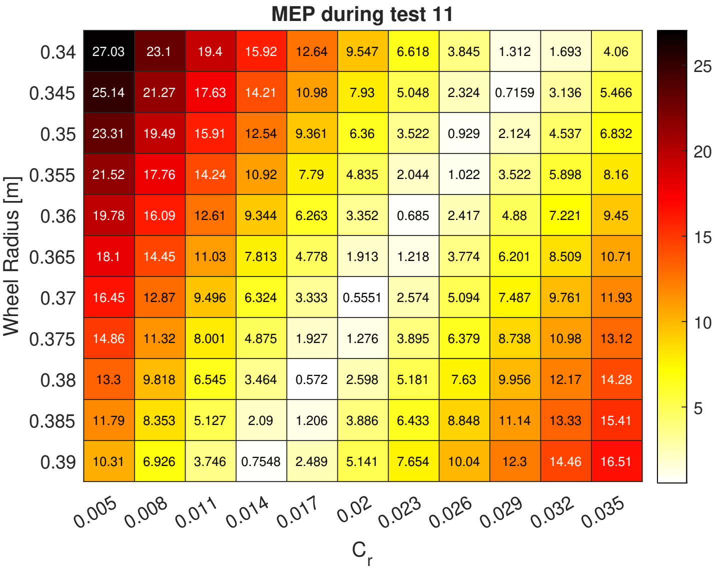


Figure C.30: *Sensitivity with multiple parameters for test 11 with multiple forgetting factors $\lambda_1 = 1$, $\lambda_2 = 0.99$.*

Bibliography

[1] Dantherm AB. URL `https://www.dantherm.com/media/2351453/hx-diagram_2016-sv.pdf`. (Accessed: 2021-06-16).

[2] Mohd Faruq Abdul Latif, Muhammed Noor Hashim, M Rashid, Muhamad Azhari, and Muhammad Nur Othman. Roof box shape streamline adaptation and the impact towards fuel consumption. volume 97, 02 2017. doi: 10.1051/matecconf/20179701089.

[3] Karl J Åström and Björn Wittenmark. *Adaptive Control*. Dover Books on Electrical Engineering. Dover Publications, 2008. ISBN 9780486462783.

[4] Dean Adam Bodenham. Adaptive estimation with change detection for streaming data. 2014.

[5] Hosam K. Fathy, Dongsoo Kang, and Jeffrey L. Stein. Online vehicle mass estimation using recursive least squares and supervisory data extraction. In *2008 American Control Conference*, pages 1842–1848, 2008. doi: 10.1109/ACC.2008.4586760.

[6] Jyotishman Ghosh, Stéphane Foulard, and Rafael Fietzek. Vehicle Mass Estimation from CAN Data and Drivetrain Torque Observer. In *SAE World Congress Experience 2017*, Mar. 2017.

[7] Fredrik Gustafsson and Svante Gunnarsson. A parametric method for modeling of time varying spectral properties. 10 1995.

[8] Erik Jonsson Holm. Vehicle Mass and Road Grade Estimation using Kalman Filter. Master's thesis, Dept. Elect. Eng., Linköpings Universitet, 2011.

[9] Narayanan Kidambi, R. L. Harne, Yuji Fujii, Gregory Pietron, and K. W. Wang. Methods in Vehicle Mass and Road Grade Estimation. In *SAE Int. J. Passeng. Cars - Mech. Syst*, volume 7, 2014. doi: 10.4271/2014-01-0111.

[10] Daeil Kim, Seibum B. Choi, and Jiwon Oh. Integrated vehicle mass estimation using longitudinal and roll dynamics. In *2012 12th International Conference on Control, Automation and Systems*, pages 862–867, 2012.

[11] Nan Lin, Changfu Zong, and Shuming Shi. The Method of Mass Estimation Considering System Error in Vehicle Longitudinal Dynamics. 2018.

[12] Bharat Mohan Redrouthu. Tyre modelling for rolling resistance. Master's thesis, Dept. Applied Mechanics., Chalmers university of technology, 2014.

[13] Rajamani Rajesh. *Vehicle Dynamics and Control*. 2006. ISBN 0-387-26396-9. doi: 10.1007/0-387-28823-6.

[14] Ayyoub Rezaeian, Reza Zarringhalam, Saber Fallah, Wael Melek, Amir Khajepour, Shih-Ken Chen, N. Moshchuck, and Bakhtiar Litkouhi. Novel Tire Force Estimation Strategy for Real-Time Implementation on Vehicle Applications. *IEEE Transactions on Vehicular Technology*, 64(6):2231–2241, 2015. doi: 10.1109/TVT.2014.2345695.

[15] Engineering ToolBox. Rolling resistance. URL `https://www.engineeringtoolbox.com/rolling-friction-resistance-d_1303.html`. (Accessed: 2021-05-20).

[16] Ardalan Vahidi, Maria Druzhinina, Anna Stefanopoulou, and Huei Peng. Simultaneous mass and time-varying grade estimation for heavy-duty vehicles. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 6, pages 4951–4956 vol.6, 2003. doi: 10.1109/ACC.2003.1242508.

[17] Ardalan Vahidi, Anna Stefanopoulou, and Huei Peng. Experiments for Online Estimation of Heavy Vehicle's Mass and Time-Varying Road Grade. 72, 01 2003. doi: 10.1115/IMECE2003-43848.

[18] Ardalan Vahidi, Anna Stefanopoulou, and Huei Peng. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: Theory and experiments. *Vehicle System Dynamics - VEH SYST DYN*, 43: 31–55, 01 2005. doi: 10.1080/00423110412331290446.

[19] Wikipedia. Automobile drag coefficient. URL `https://en.wikipedia.org/wiki/Automobile_drag_coefficient#Drag_area`. (Accessed: 2021-06-10).

[20] Robert Wragge-Morley, Guido Herrmann, and Stuart Burgess. Gradient and Mass Estimation from CAN based data for a light passenger car. In *SAE International Journal of Passenger Cars - Mechanical Systems*, 2015.