
Automotive Power Measurement

with Rotation, GPS and On-Board-Diagnostics Sensor Data

Project Electrical Engineering AS2019

Authors

Julian Bärtschi, Michael Schmid

Supervisor

Prof. Dr. Guido Schuster

Assistant Supervisor

Reto Flütsch, Insoric AG

Subject

Digital Signal Processing / Statistical Machine Learning

Abstract

Abstract

The project is about power measurement device for cars, named *Insoric RealPower* from the Swiss company Insoric AG. The measurement device was released about nine years ago and is well established. The key module is a rotation sensor which is fixated in the center of the wheel. With this frequency and the rolling circumference of the wheel, the included software is able to compute the velocity and the corresponding power of the car.

First Problem: Rolling Circumference

The development team is currently working on a GPS module as an alternative method to record the cars velocity. Measurements have shown that the velocity of these two measurements systems vary slightly. It is supposed that this error occurs because the rolling circumference is falsely measured. A first Task is to investigate this problem and derive a method to compute the circumference by using the data from the GPS and RealPower system. An additional purpose of this task is to deliver an analytical approach to show how an error in the speed measurement propagates through the whole process to the end result the power.

We were able to write a Python script which processes the Data from the two systems and computes the rolling circumference of the wheel. Because the two systems have a different sampling rate and no timestamp the data had to be synchronized. To find the delay between the two system, the data had to be interpolated and correlated. For the computation of the circumference various filter methods and linear regression models have been created.

Second Problem: Wheel to Motor Ratio

To compute not only the power but also the widely used torque of the motor, the rotational speed of the motor is needed. Currently the user has to deliver the ratios between the wheel and the motor per gear. Either he/she knows the ratios or has to measure it. For that procedure, the speed has to be measured by driving with a steady rotational speed of the motor. Afterwards the ratio is computed in the program by selecting this steady sequence.

The idea is to compute the ratios for each gear with the cars On-Board-Diagnostic System (OBD), OBD is a protocol which allows to read a wide variate of data from the car, including the velocity and rotational speed of the motor. This Task includes buying a device and find a way to process the data and calculate the ratios. Two different devices have been purchased. The rotational speed as an example, can be read sufficiently fast. The communication is ensured over Bluetooth and can be controlled with a Python script. With an estimation maximization algorithm the ratio can be approximated with a Gaussian distribution.

Contents

Contents

Abstract	iii
1 Introduction	1
2 Fundamentals and used technology	3
2.1 Fundamentals of physics	3
2.1.1 Power	3
2.1.2 Torque	4
2.1.3 Coasting and its differential equation	4
2.2 On Board Diagnostics	7
2.2.1 Message structure	7
2.2.2 CANedge1	8
2.2.3 OBDLink MX+ and ELM327	8
2.3 RealPower Module	9
2.3.1 Velocity	10
2.3.2 Acceleration	10
2.3.3 Drag and Rolling Resistance Coefficient	10
2.4 GPS Module	11
2.5 Savitzky Golay	12
3 Methods	13
3.1 Rolling Circumference	13
3.1.1 Time delay with the Correlation	13
3.1.2 Time Delay with the Mean Error Squared	14
3.1.3 Computation of the Rolling Circumference	16
3.1.4 Influence of a Incorrectly Inflated Tire	18
3.1.5 Comparison to the Intrinsic Wheel Measuring Device	19
3.1.6 Velocity and the Circumference	19
3.2 Error Analysis	20
3.2.1 Analytical Approach	20
3.2.2 Numerical Approach	22
3.3 Transmission Gear Ratio	22
3.3.1 Delay by Histogram Matching	23
3.3.2 Gaussian Mixture Model	23
4 Recommendations	27
4.1 Initial drive	27

Contents

5 Conclusion	29
---------------------	-----------

Chapter 1

Introduction

The Institute for Communication Systems ICOM at the HSR has developed together with the Insoric AG the RealPower System. Which measures the rotational frequency of the wheel. Through the application of physics the power can be computed. Furthermore exists an other ICOM developed RealPower systems prototype, which works over GPS. A challenge is now, that the measured velocity of the GPS Module not exactly with the velocity of the rotational system matches. The speed of the rotational system is computed with the frequency of the wheel and its circumference. The diameter is measured by the user with a specialized caliper. The assumption for this difference in speed is that the diameter of the wheel is not measured correctly.

Another challenge is that the system needs the rotational speed of the motor to derive the torque from the power of the car. The rotational speed of the motor is currently computed through the ratio between the rotational speed of the wheel and the speed. If those ratios are unknown, the user has to drive with an steady speed and remember its belonging rotational speed. The RealPower system can compute the needed ratios, with the velocity from the rotation system and the rotational speed of the motor. This method is not only unpractical but also quiet dangerous. Modern cars are capable of delivering the rotational speed and other information about the car over an protocol called OBD-II (On Board Diagnostics).

The aim of this project is to resolve the problem with the difference in speed between the two measurement systems. A reliable method to estimate the rolling circumference has to be focused. It also has to be quantified, how much a falsely measured speed influences the computation of the power and its result. Furthermore has a suitable OBD-II system to be evaluated and purchased. This should allow us to know the rotational speed at any time and to identify the current transmission gear.

Chapter 2

Fundamentals and used technology

2.1 Fundamentals of physics

Because of the main topic of our project, the power measurement of a car, a strong theoretical knowledge in physics is therefore inevitable. In this chapter the background for computing the power will be presented.

2.1.1 Power

The power itself is defined as

$$P = F v \quad (2.1)$$

For the application with the car we need the total force which the car has to deal with and its speed.

For the acceleration force can Newtons second law be used, "*the vector sum of the forces on an object is equal to the mass of that object multiplied by the acceleration of the object*" [1].

$$F_a = m a \quad (2.2)$$

While driving a car has to overcome further forces [2]:

The drag

$$F_D = \frac{1}{2} \rho v^2 C_D A \quad (2.3)$$

the rolling resistance

$$F_R = C_R m g \quad (2.4)$$

the gradient resistance

$$F_G = m g \sin(\alpha) \quad (2.5)$$

and the translational acceleration resistance of all spinning parts. Where J_R the moment of inertia, r the radius of the spinning part, and R the dynamic wheel diameter with slip

$$F_T = a \left(m + \sum_{j=1}^n \frac{J_{Rj}}{r_j R_j} \right) \quad (2.6)$$

For the applied computation of the power, the part of the translational acceleration resistance is neglected. It is just not applicable to get all the information about the internal part of the car. The combined formula for the power can now be written as

$$P = (F_A + F_D + F_R + F_G) v \quad (2.7a)$$

$$P = \left(m a + \frac{1}{2} \rho v^2 C_D A + C_R m g + m g \sin(\alpha) \right) v \quad (2.7b)$$

2.1.2 Torque

Beside the power, the torque M is also a well established unit in car manufacturing.

$$P = M \omega(v) \quad (2.8)$$

Either you compute the torque from the wheel or the motor. For each torque the corresponding rotational speed ω has to be plugged in the equation.

Let's rewrite the power with (2.1) or written out with (2.7).

$$M \omega(v) = F(v) v \quad (2.9)$$

Writing ω as the frequency

$$M 2\pi f(v) = F(v) v \quad (2.10)$$

With the circumference C of the wheel

$$M 2\pi \frac{v}{C} = F(v) v \quad (2.11a)$$

$$M 2\pi \frac{\psi}{C} = F(v) \psi \quad (2.11b)$$

The torque is defied for the application with a car.

$$M = \frac{F(v) C}{2\pi} \quad (2.12)$$

2.1.3 Coasting and its differential equation

For the drag force in (2.3) is the coefficient C_D and fore the rolling resistance in (2.4) is the coefficient C_R needed. To get these coefficients from a single measurement with the car, the user has to coast to a stop. It is referable if the test is performed on a flat street, because the gradient resistance has no influence.

If the car is coasting only two forces are present, the drag- and rolling force.

$$F_{tot} = F_D + F_R \quad (2.13a)$$

$$m a = -\frac{1}{2} \rho v^2 C_D A - C_R m g \quad (2.13b)$$

Because the acceleration while coasting is negative, the forces are also negative. The term a and v can be rewritten as derivatives from the position x .

$$m \ddot{x} = -\frac{1}{2} \rho C_D A \dot{x}^2 - C_R m g \quad (2.14)$$

The equation in (2.14) is second order ordinary differential equation for the coasting of a car. It is rather difficult to solve this equation. An other approach is to take an solution which can be found online and prof that it is a correct one. Therefore a few substitution are helpful.

$$B = \frac{1}{2} \rho C_D A \quad (2.15)$$

$$\ddot{x} = -B \dot{x}^2 - C_R g \quad (2.16)$$

$$y = \dot{x} \quad (2.17)$$

$$\dot{y} = -B y^2 - C_R g \quad (2.18)$$

Here the solution for y from a great article on the website fondam [3]

$$y = \sqrt{\frac{C_R g}{B}} \tan\left(\sqrt{\frac{C_R g}{B}} (k_1 - B t)\right) \quad (2.19)$$

The integration constant k_1 is known from the initial velocity and is a constant.

$$y(t=0) = v_0 \Rightarrow k_1 = \frac{1}{W} \arctan\left(\frac{v_0}{W}\right) \quad (2.20)$$

With another substitution $W = \sqrt{\frac{C_R g}{B}}$ we can further simplify the equation

$$y = W \tan(W k_1 - W B t) \quad (2.21)$$

The first derivative of the solution is equal to (2.18)

$$\dot{y} = -W^2 B \sec^2(W k_1 - W B t) \quad (2.22a)$$

$$-B y^2 - C_R g = -W^2 B \sec^2(W k_1 - W B t) \quad (2.22b)$$

For the term y^2 the solution can be inserted. With some algebra this equation is turning out as true.

$$-W^2 B \sec^2(W k_1 - W B t) = -B (W \tan(W k_1 - W B t))^2 - C_R g \quad (2.23a)$$

$$-W^2 B \sec^2(W k_1 - W B t) = -B W^2 (\sec^2(W k_1 - W B t) - 1) - C_R g \quad (2.23b)$$

$$-W^2 B \sec^2(W k_1 - W B t) = -B W^2 \sec^2(W k_1 - W B t) + B W^2 - C_R g \quad (2.23c)$$

$$0 = B W^2 - C_R g \quad (2.23d)$$

$$C_R g = B W^2 \quad (2.23e)$$

$$C_R g = \frac{\rho C_W A}{2} \left(\sqrt{\frac{C_R g}{B}} \right)^2 \quad (2.23f)$$

$$C_R g = \frac{\rho C_W A}{2} \left(\sqrt{\frac{C_R g}{\frac{\rho C_W A}{2}}} \right)^2 \quad (2.23g)$$

$$C_R g = C_R g \quad (2.23h)$$

The solution for the velocity y is further used for the definition of the two coefficients C_R and C_D .

2.2 On Board Diagnostics

In order to measure the revolutions per minute (rpm) of the motor, there is the so-called on-board diagnostic system (OBD). The on-board diagnostic system is implemented in all modern automotives. It's intended to give important information to the vehicle owner or a repair technician.

The first self-scanning System was implemented by Volkswagen in the year 1969. Later on, the OBD-I regulation was supposed to encourage car manufacturers to design a reliable emission control system. However, OBD-I had the problem that it was not standardized. Thus, it was not really successful. In the year 1996 the OBD-II standard was made mandatory for all cars sold in the United States of America. In 2001 and 2004 the European on-board diagnostics (EOBD) regulations - which are the European representative to the OBD-II - have been declared for petrol and diesel cars respectively. [4]

The on-board diagnostic system is capable of sending different kinds of sensor data through a vehicle-bus such as the Controller Area Network (CAN). A well-known example is the “check engine light”(figure 2.1), which indicates, that there is some kind of malfunction with the engine. Not only is it possible to read out error messages etc. but also to send commands to the vehicle. An important command is the request for different kinds of sensor data. The access of the OBD-II Data takes place via the diagnostic connector of the car, which can be seen in figure It is required to be in the cars interior and within 2 feet of the steering wheel.

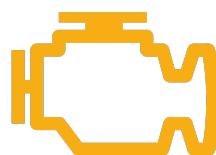


Figure 2.1: The check engine light (also called malfunction indicator lamp (MIL)) indicates a malfunction of the vehicle. <https://commons.wikimedia.org/wiki/File:Motorkontrollleuchte.svg>

2.2.1 Message structure

Since all of the cars signals get transmitted through the vehicle-bus the OBD-System sends Data only after it gets requested. In order to do that, the Society of Automotive Engineers (SAE) has defined the standard J1979, which contains so-called Parameter IDs (PIDs). These Parameter IDs identify the available sensors of which data can be requested from. In this standart a set of mandatory PIDs is defined, which every car needs to support. However, many car manufacturers provide additional IDs. [5]

Every request, as well as the corresponding response are structured in a 75-bit-long message, as it can be seen in figure 2.2. The first 11 bits of the message are called the Identifier and the following 64 bits are for the data. A Request has the identifier 7DF and the Response has identifiers ranging from 7E8 to 7EF. The first 2 hexadecimal values contain information about the length in number of bytes of the remaining data. Unused bytes get padded with either 5, for requests, or 'a' for responses. There are 4 different modes available to the user. In order to request current data the Mode 01 is needed. Next, there are 2 bytes for the PIDs.

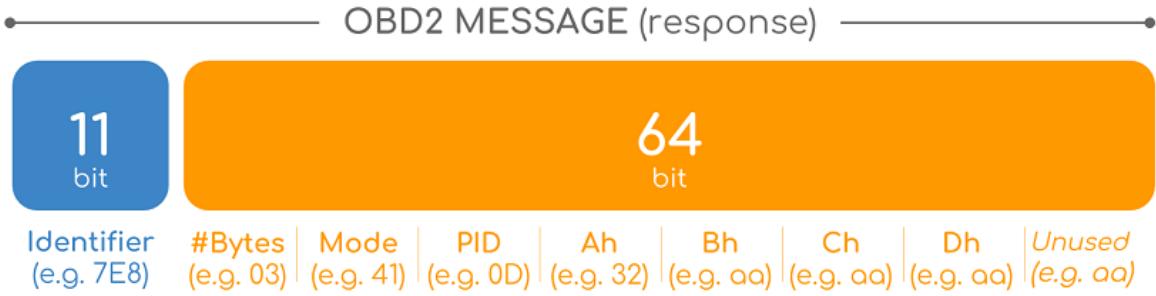


Figure 2.2: An OBD message is structured in a 75-bit-long bitstream containing a 11-bit-long Identifier followed by the data (64 bit). [5]

As an example, to get the current vehicle speed, the request

`7DF 02 01 0D 55 55 55 55 55`

gets sent. Note, that the first 11 bit indicate a request. The following byte with the value 02 indicate, that there are 2 bytes of data left. These contain the mode and Parameter ID, where the mode 01 means "Show current data" and 0D is the identifier for Vehicle speed.

In this example, as it can be seen in figure [5], the response

`7E8 03 41 0D 32 aa aa aa aa`

contains the requested data. Similar to the request, the message contains Identifier, number of following bytes, the mode and the according PID. Additionally, the message also contains 1 more byte of data, the value. According to this message, the vehicle travels at a speed of 50km/h (32 hexadecimal equals 50 decimal).

2.2.2 CANedge1

As a first attempt to log the cars rpm the CANedge1 from CSSElectronics was used. The purpose of the CANedge1 is to log CAN data. Via a RS-232 to OBD connector it can be plugged into the vehicle. It starts logging as soon as it gets a connection. In advance to the use, a "config-file" needs to be modified in order to reach the desired behavior. It turned out that this device works really well, given that the config-file is modified correctly. Unfortunately, it did not work with all the tested cars. As example, it worked as desired with cars of the Volkswagen Group (Audi, VW, Škoda) but it didn't work with brands like Fiat or Opel.

It could not have been investigated extensively, if this fault lies on a wrong application of the device or if these car models do not support this kind of requests or devices.

2.2.3 OBDLink MX+ and ELM327

For a second attempt the MX+ from OBDLink® came to use. The MX+ is a Bluetooth OBD adapter which can be connected to Notebooks, Smartphones etc. Note, in order to request and receive Data the device must be able to transmit as well as receive Bluetooth Data.

Like many OBD connectors the MX+ can be accessed via the ELM327 interface. As a widely used interface the ELM327 from the company ELM Electronics can be found in many low-cost devices. Due to the python-OBD library easy access is enabled.

```

1 import numpy as np
2 import obd
3
4 # Setup the debug mode to display everything in the console
5 obd.logger.setLevel(obd.logging.DEBUG)
6 # return list of valid USB or RF ports
7 ports = obd.scan_serial()
8 # my laptop works with port COM5
9 connection = obd.OBD(r"\COM5")
10 # select an OBD command speed (sensor)
11 cmd_speed = obd.commands.SPEED
12 # select an OBD command RPM (sensor)
13 cmd_RPM = obd.commands.RPM
14
15 # print true if connected successful
16 print(connection.is_connected())
17
18 # initialize lists and counting variable i
19 time_rpm = []
20 time_speed = []
21 rpm = []
22 speed = []
23 i = 0
24
25 # measure the data and store it after 1000 measured values
26 if connection.is_connected() is True:
27     try:
28         i = 0
29         while(1):
30             sresponse = connection.query(cmd_RPM)
31             speed_3 = connection.query(cmd_speed)
32
33             rpm.append(float(sresponse.value.magnitude))
34             speed.append(float(speed_3.value.magnitude))
35             time_speed.append(speed_3.time)
36             time_rpm.append(sresponse.time)
37             i += 1
38             if i >= 1000:
39                 data = np.c_[time_rpm, rpm, time_speed, speed]
40                 with open("output.txt", "a") as f:
41                     np.savetxt(f, data, delimiter=';')
42                 time_rpm = []
43                 time_speed = []
44                 rpm = []
45                 speed = []
46                 i = 0
47 # ctrl+C in the console to terminate the program
48     except KeyboardInterrupt:
49         pass
50
51 data = np.c_[time_rpm, rpm, time_speed, speed]
52
53 with open("output.txt", "a") as f:
54     np.savetxt(f, data, delimiter=';')

```

2.3 RealPower Module

The RealPower system allows to measure the power of a car by driving on the street. No expensive chassis dynamometer is required.

How to compute the power of the car with only the RealPower module, which measures the rotational speed of the wheel, will be described in this section of the paper.

As in section 2.1 described allows physics to compute power if following parameters are known.

- Velocity
- Mass
- Drag coefficient
- Rolling coefficient
- Temperature and air-pressure

With all this values the equation for the power (2.7) can be applied. The user has to perform the test run on a more and less flat street. With this the gradient force can be neglected. The user

has to know the mass of the car under test. With the temperature T , the air-pressure p and the specific gas constant for dry air R , the density ρ of the air can be computed.

$$\rho = \frac{P}{R T} \quad (2.24)$$

2.3.1 Velocity

The Velocity is measured by the RealPower module an rotational sensor. The sensor is fixed on the rotational axis of the wheel by a double-faced adhesive tape.



Figure 2.3: The RealPower Module [6]

The Module measures the rotational speed or in other terms the frequency f of the wheel in hertz. The speed can be computed with the rolling circumference C of the wheel.

$$v = C f \quad (2.25)$$

2.3.2 Acceleration

The acceleration of the car must be known during the acceleration and coasting test. The speed curve of these two test are measured by the module. Afterward a polynomial 3rd order can be fitted on this curve.

$$v = \hat{\beta}_3 t^3 + \hat{\beta}_2 t^2 + \hat{\beta}_1 t + \hat{\beta}_0 \quad (2.26)$$

The acceleration follows by the derivation of the speed.

$$a = \dot{v} = 3\hat{\beta}_3 t^2 + 2\hat{\beta}_2 t + \hat{\beta}_1 \quad (2.27)$$

2.3.3 Drag and Rolling Resistance Coefficient

To compute the drag and rolling resistance coefficient, a coasting run on a flat street has to be performed. There are two possibilities to compute the coefficients, either by fitting the coasting curve to a polynomial or to the solution of the differential equation (2.19).

Polynomial approach

Rewriting the equation (2.13b)

$$m a = -\frac{1}{2} \rho v^2 C_D A - C_R m g \quad (2.28a)$$

$$-a = \left(\frac{\rho v^2}{2 m} \right) C_D A + g C_R \quad (2.28b)$$

This equation exists for n datapoints in the coasting curve and has two unknowns $C_D A$ and C_R .

$$\mathbf{A}x = b \quad (2.29)$$

This problem can be solved with Gauss's Least Squares approximation.

$$x = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T b \quad (2.30)$$

$$\mathbf{A} = \begin{bmatrix} \frac{\rho v_0^2}{2 m} & g \\ \vdots & \vdots \\ \frac{\rho v_n^2}{2 m} & g \end{bmatrix}, \quad b = \begin{bmatrix} -a_0 \\ \vdots \\ -a_n \end{bmatrix}, \quad x = \begin{bmatrix} C_D A \\ C_R \end{bmatrix} \quad (2.31)$$

Differential equation

Because the solution from the differential equation is not so simple as the polynomial fit, a Python library is used for fitting the curve. The function `curve-fit` from the library `scipy` can be used. The parameter `p0` stands for the initial values which are to be fitted. In figure 2.3.3 the code example

```

1 def curvefit_cd_cr(t, v, mass):
2
3     def func(t, cd, cr):
4         g = 9.81
5         v0 = v[0]
6         rho = 1.2041
7         B = rho*cd/(2*mass)
8         W = np.sqrt(cr*g/B)
9         k1 = 1/W*np.arctan(v0/W)
10        v = W*np.tan(W*(k1-B*t))
11        return v
12
13 [cd, cr], pcov = curve_fit(func, t, v, p0=[0.67, 0.005])

```

An example in 2.4 of the curve fitting function on a coasting test run.

2.4 GPS Module

As an addition to the rotation sensor, the Insoric AG is developing a GPS module. With this, no sensor has to be placed at the wheel. The `ublox` chip is delivering the speed and it would also give the gradient angle which the car is driving. For long term thinking a sensor which delivers the gradient angle would be very helpful. The test runs would not have to be performed on a flat street.

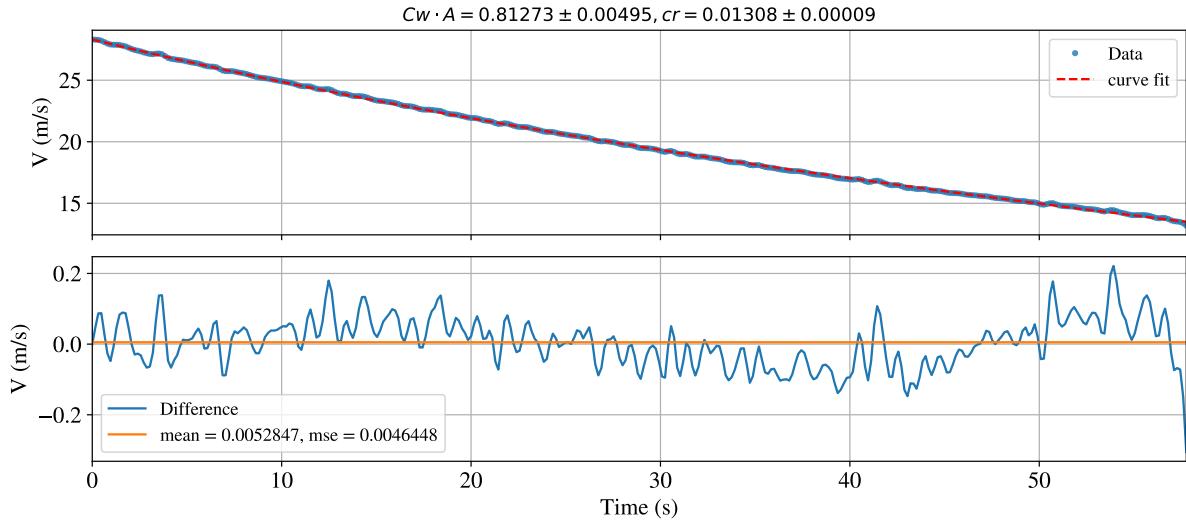


Figure 2.4: Differential equation fitted to a coasting test run

2.5 Savitzky Golay

In several occasions the Savitzky Golay smoothing filter is used. [7] The parameters for the filter are the polyorder n and the window length M . The basic idea of the filter is that it takes M elements of the data m and fits a polynomial with order n in it.

$$\hat{y}_m = \hat{\beta}_0 + \hat{\beta}_1 m + \hat{\beta}_2 m^2 + \hat{\beta}_3 m^3 + \cdots + \hat{\beta}_n m^n \quad (2.32)$$

With the same approach as in 2.3.3 and equation (2.30), with the matrices

$$\mathbf{A} = \begin{bmatrix} 1_0 & m_0 & m_0^2 & \dots & m_0^n \\ 1_1 & m_1 & m_1^2 & \dots & m_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1_M & m_M & m_M^2 & \dots & m_M^n \end{bmatrix}, \quad b = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_M \end{bmatrix}, \quad x = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_M \end{bmatrix} \quad (2.33)$$

The derivation can also be estimated by this filtering method.

$$\dot{\hat{y}}_m = \hat{\beta}_1 + 2\hat{\beta}_2 m + 3\hat{\beta}_3 m^2 + \cdots + n\hat{\beta}_n m^{n-1} \quad (2.34)$$

or the second derivative

$$\ddot{\hat{y}}_m = 2\hat{\beta}_2 + 6\hat{\beta}_3 m + \cdots + n(n-1)\hat{\beta}_n m^{n-2} \quad (2.35)$$

Chapter 3

Methods

3.1 Rolling Circumference

How in section 2.3 described is the rolling circumference of a car a very important parameter to compute the speed of a car. With the new GPS module and the current RealPower module are two different measurement systems available to measure the speed, or the correlated frequency of the wheel. From the datasheet of the GPS chip is known that the measurement is reliable with an accuracy from 0.1 m/s. Initial tests have also shown that especially at higher speed and a good signal the measurements can be trusted.

3.1.1 Time delay with the Correlation

The two modules are not able to deliver a time stamp. When the device is started, the time is set to zero and will count up with the given sample time. A first task to compute the rolling circumference is to align the two signals. The correlation can be used for this task. While correlating the two signals a peak in the correlation value should appear when the two signal are perfectly aligned. In the equation for the correlation (3.1) can you see that the two discrete signals have to be at the same datapoint m .

$$R_{fg}[n] = (f * g)[n] = \sum_{m=0}^{\infty} f[m] g[m + n] \quad (3.1)$$

Because the two measurement modules have a different sampling frequency, at least one signal has to be interpolated. In figure 3.1 the process to correct the time shift between the two signals is shown. The time base for the interpolation gives always the shorter signal. In this example the GPS signal. The correlation in the second plot is given by

$$(v_{gps} * u_{rp})[n]. \quad (3.2)$$

With the time index of the peak value of the correlation the delay τ can with

$$\tau = \max\{t_{RealPower}\} - \text{argmax}\{correlation\} \quad (3.3)$$

be calculated.

To calculate the time delay with the correlation has one major problem: If the two signal are very different. In example one signal is only a short part of the other and has a part with higher average velocity than the shorter one. The correlation will detect a wrong peak.

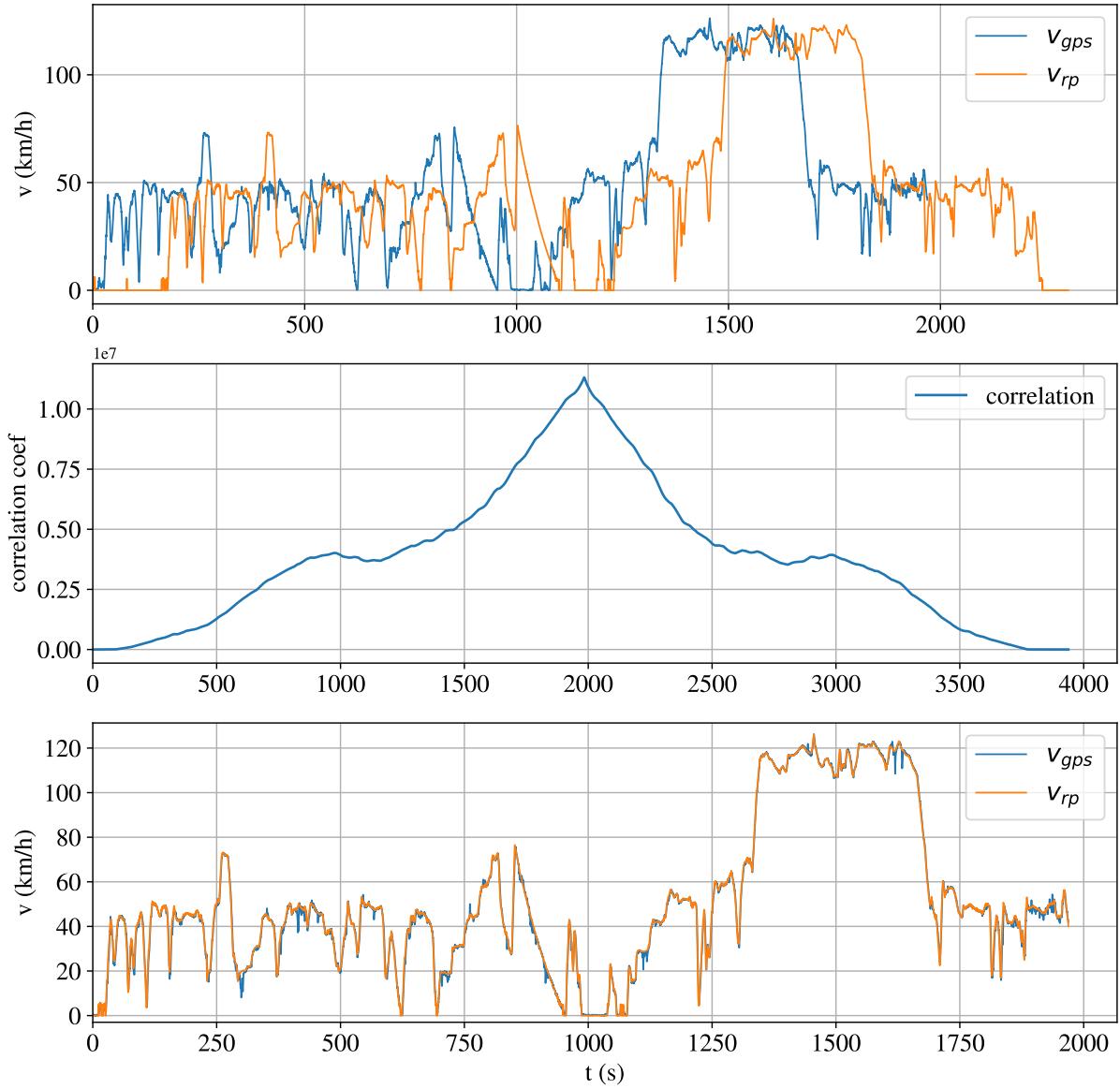


Figure 3.1: Correlation

As you in 3.2 can see, the part on the highway, which is about 120 km/h will generate peaks, when the other signal is also quite high. The correct peak would be the first one and the highest.

3.1.2 Time Delay with the Mean Error Squared

The delay can be computed without an correlation with an different approach. The idea is that the shorter signal will be compared with a windows of the longer signal with the same length. The short signal s will be normalized, as well every window of the longer signal l_i . The mean squared error indicates how well the signals are aligned.

$$\text{MSE}_i = \frac{1}{n_s} \sum_{j=0}^{n_s} (s_j - l_{j,i})^2 \quad (3.4)$$

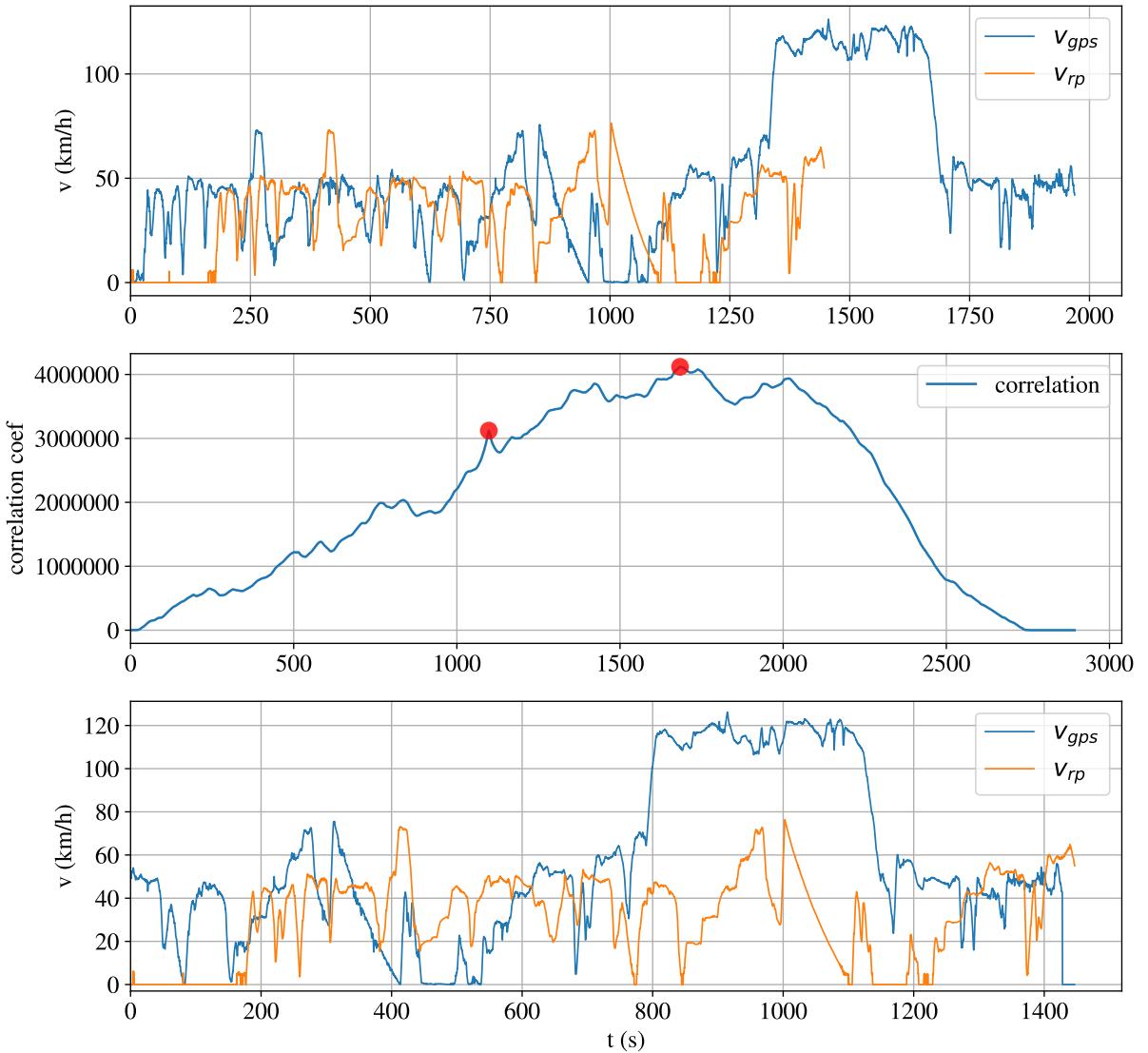


Figure 3.2: Failed correlation

In figure 3.3 the measurement with the RealPower module is shorter and has been started earlier. That the correct window of the GPS signal is selected, zeros must be added before the signal starts. It is not possible to know in general if the shorter signal started earlier or stopped later, therefore zero padding is applied on both sides of the signal.

With the time index of the minimal value of the MSE and the zeropadded time t_{zero} , the delay τ can with

$$\tau = \text{argmin}\{\text{MSE}\} - t_{zero} \quad (3.5)$$

be calculated.

The computational resources for this kind of sum operations are rather high. To reduce the resources the MSE is computed two times. For the first time the window is shifted for 100 datapoints. The minimum MSE is still clearly noticeable but is not that accurate, the index of this minimal value is stored in i_{min} . In the second loop, the window is shifted by one datapoint. The first window starts at the index i_{min} minus 200 and ends at i_{min} plus 200.

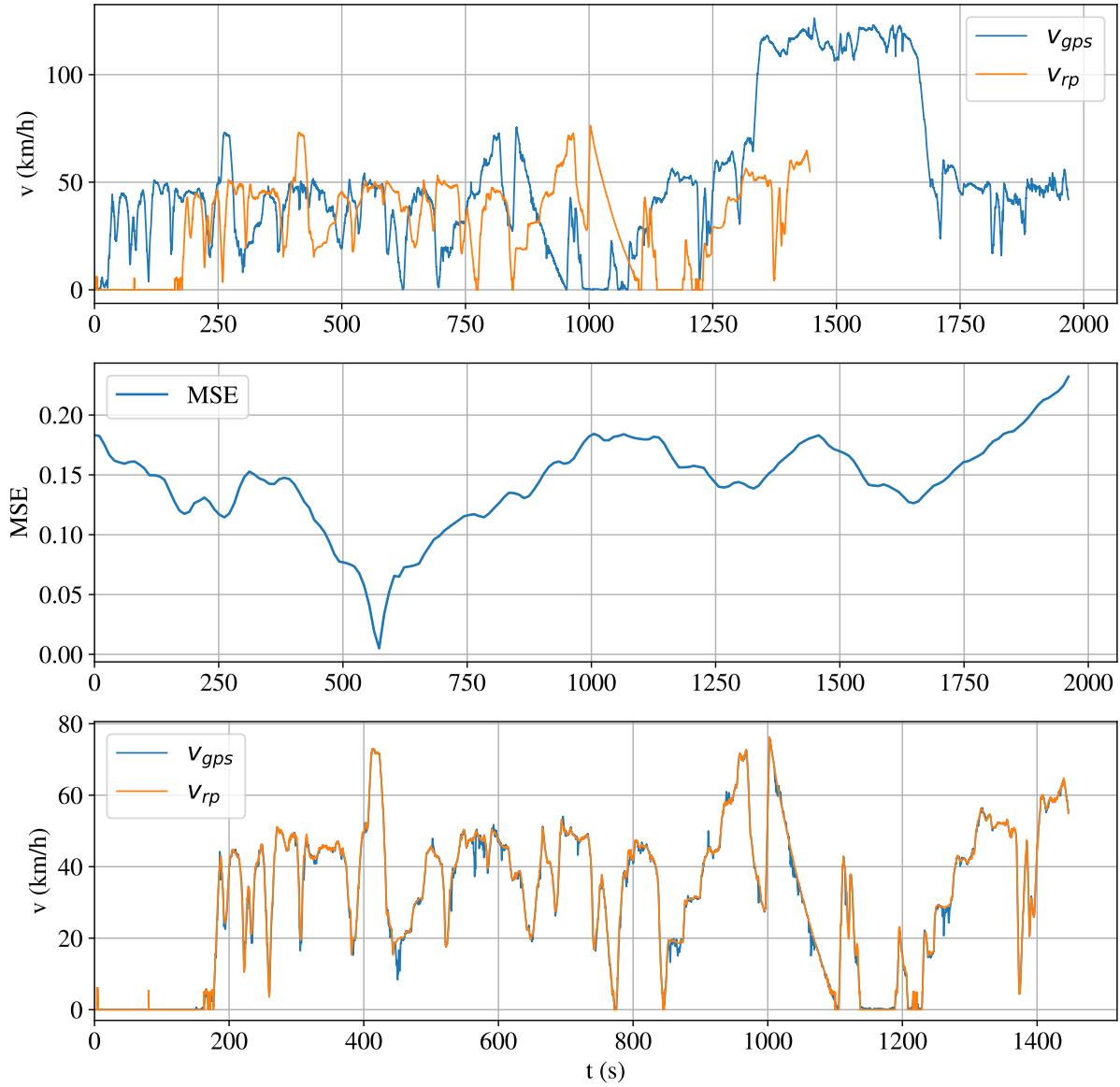


Figure 3.3: Delay with MSE

3.1.3 Computation of the Rolling Circumference

The computation can be implemented in various ways, but the formula which has to be used is always the same.

$$C = \frac{v_{car}(t)}{f_{wheel}(t)} \quad (3.6)$$

Point by Point Division

One method is to divide every datapoint from the GPS signal by the corresponding point from the RealPower signal. The processing of the data can be done afterwards. Because the frequency is sometimes zero and is in the denominator, a zero division occurs frequently. This data can be ignored easily. More problematic is when the car is driving very slow. If the frequency in the

denominator is slightly off, the error for the circumference rises very fast. In addition to this, is the GPS gets quite inaccurate while driving slow. Because of this problem, the circumference is only calculated when the car was driving a certain speed.

Therefore a list with every computed circumferences can be generated and be sorted after the corresponding speed. The *Nan* results can with this step also be deleted.

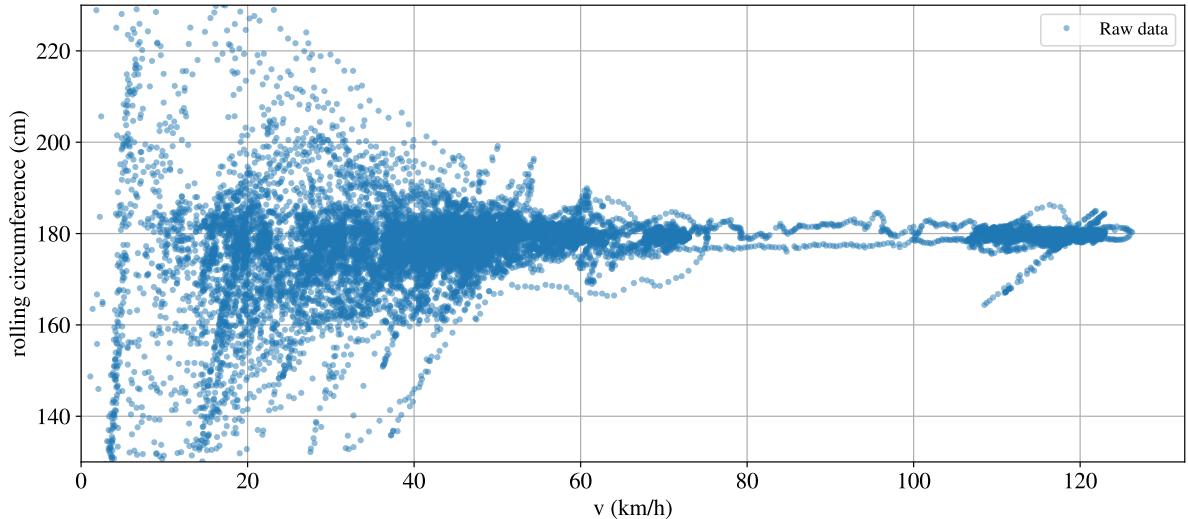


Figure 3.4: Circumference sorted by speed

As you in 3.4 can see the variance is rather big. Especially while driving slower. This data can further be used to compute the effective circumference. There are several possible approaches.

- Averaging the data bigger than an certain speed.
- Fit a straight line $C = \hat{\beta}_1 v + \hat{\beta}_0$ in the data.
- Fit a polynomial $C = \hat{\beta}_2 v^2 + \hat{\beta}_1 v + \hat{\beta}_0$ in the data.
- Filter the data with a Savitzky Golay Filter.

Filtering the data would reduce the variance. But the main problem, the division, has been implemented long ago.

Averaging before Division

Every system has some noise η in it. In general the noise is normal distributed.

$$C(t) = \frac{V_{GPS}(t) + \eta_{GPS}(t)}{f_{wheel}(t) + \eta_{wheel}(t)} \quad (3.7)$$

The division with the noise is not optimal. It can be avoided by averaging before the division.

$$C(t) = \frac{\sum_{n=0}^N [v_{GPS}(t) + \eta_{GPS}(t)]}{\sum_{n=0}^N [f_{wheel}(t) + \eta_{wheel}(t)]} \quad (3.8)$$

The advantage with averaging is that the noise has an expectation value of zero: The noise should theoretically disappear.

$$E [\eta_{GPS}(t)] = 0 \quad (3.9)$$

$$E [\eta_{Rotation}(t)] = 0 \quad (3.10)$$

For picking the right data to average, a certain speed limit has to be passed. In addition can the data also be sorted after its first derivative.

It has been observed, that either the GPS or the RealPower system is not very accurate while accelerating or hitting the breaks.

The derivative of the GPS-signal is taken by the Savitzky Golay Filter. A good threshold for the speed is about $v \geq 50 \frac{\text{km}}{\text{h}}$ and for the acceleration about $a \leq 0.15 \frac{\text{m}}{\text{s}^2}$. In figure 3.5 the linear

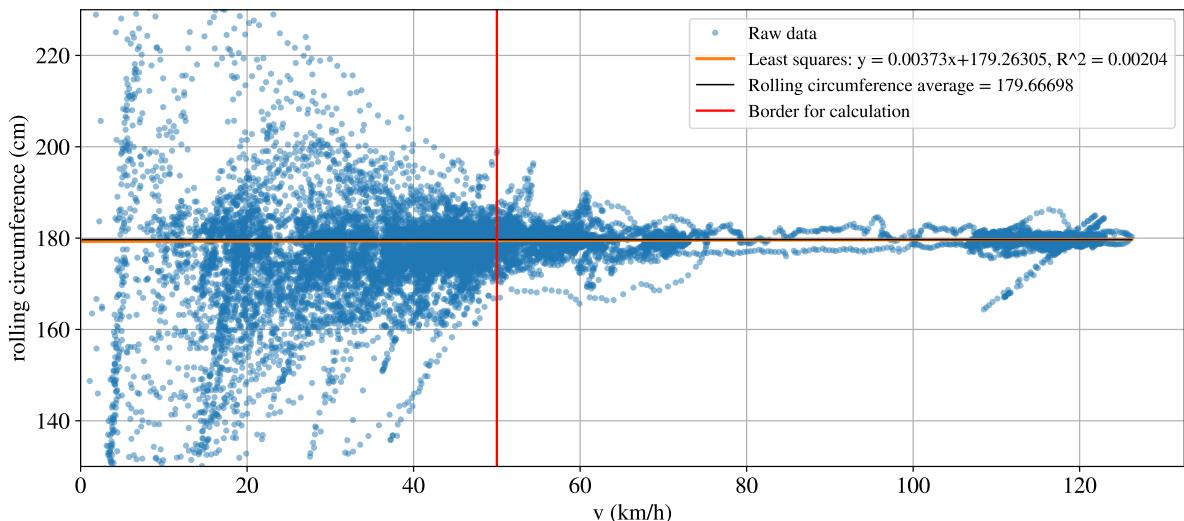


Figure 3.5: Circumference with averaging and linear regression

regression model and the circumference by averaging is plotted.

3.1.4 Influence of a Incorrectly Inflated Tire

The Insoric AG suggested to the user of the system to pump the tires according the given specifications. It is believed that a tire which is not properly pumped, has slightly smaller diameter. This hypothesis can with this new method to compute the circumference be accepted or rejected.

The test runs has been performed with an Opel Combo. The tires had been inflated with fix installed compressor at a gas station. The pressure measuring device, has not been calibrated or tested. But it can be said, that there has been a different pressure between the test runs. The diameter in the second column was calculated by the averaging method.

The variance of the averaging method can not be computed because the division between two averaged signals is taken. Following the probability density function of the with Rotation, GPS and On-Board-Diagnostics Sensor Dataoint to point computation of the circumference. It can not be directly compared with the circumference by the averaging method but it is still an indicator for the variance.

Pressure	Circ./Dia. computed	Dia. horizontal mes.	Dia. vertical mes.
3 bar	194.2570 cm / 61.834 cm	63.8 cm	61.95 cm
2.5 bar	194.0636 cm / 61.772 cm	64 cm	61.95 cm
1.5 bar	193.1910 cm / 61.495 cm	64.5 cm	61.25 cm

Figure 3.6: Circumference measurement with different wheel pressure

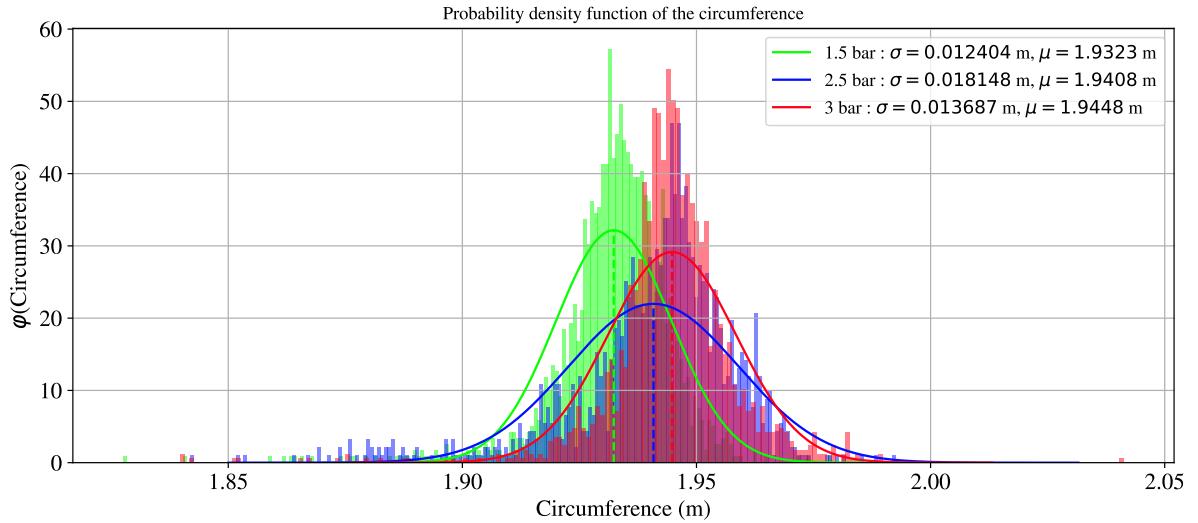


Figure 3.7: Probability density function of the circumference

3.1.5 Comparison to the Insoric Wheel Measuring Device

In 3.6 the diameter measured with the Insoric wheel measuring device is displayed. Measured horizontal means that the caliper gauge is applied horizontal to the wheel. Whereas with the vertical measurements the gauge has been applied between the floor and the top of the wheel. If the diameter computed with the written program is assumed correctly, measuring the wheel's diameter should be performed with the vertical method. This may be intuitively not the logical choice, because the wheel is vertically quite deformed whilst the car is standing still. But it seems that the diameter while driving is pretty close to the vertical measurement.

3.1.6 Velocity and the Circumference

Another question in with the circumference is if the velocity does effect the circumference of the wheel. As in 3.1.3 described, before computing the circumference the car should drive at least a certain speed. In general 50 km/h is a reasonable choice. Now what happens to the circumference if this threshold value is changed.

This plot is produced with the same data as in 3.1. Until a certain threshold the circumference varies only a little. Between 50 and 110 km/h the circumference varies in the range of 179.66 and 179.75 cm, this results in a diameter range of 0.286 mm, this can clearly be neglected.

But the circumference or also the diameter for that matter is increasing steadily. More measurements especially some with higher speeds are to be performed. Due to the speed regulations

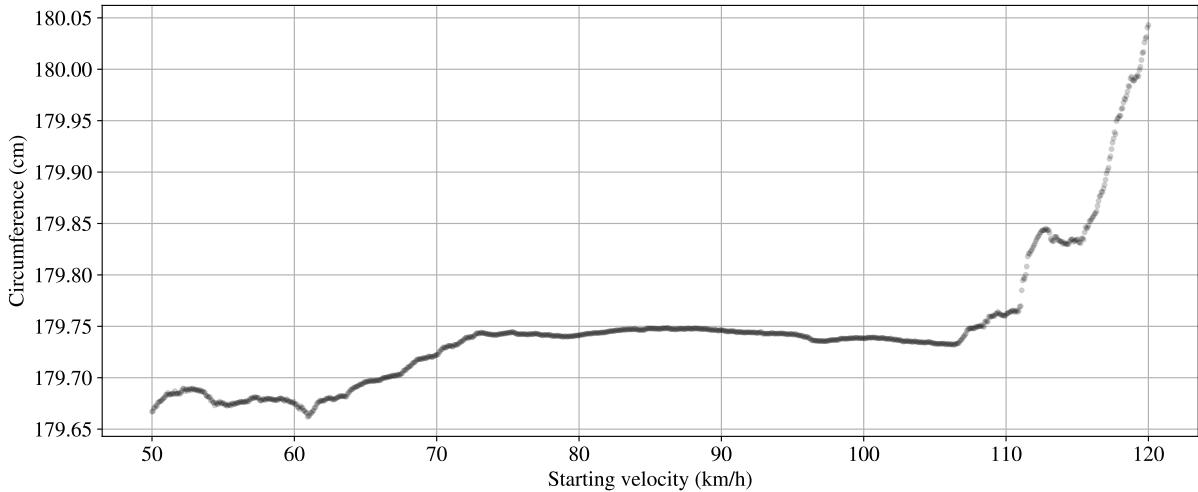


Figure 3.8: Circumference computed with a different minimum threshold for the velocity

in Switzerland, no measurements above 120 km/h has been performed.

3.2 Error Analysis

The previous section was all about computing the right rolling circumference of the wheel. The circumference is currently measured by a special caliper gauge. An imprecisely measured value, will influence the speed, and therefore the power. In this section the task is to show, how much has a falsely measured speed influence of the computed power.

3.2.1 Analytical Approach

The parameter which defines a wrong speed is called α and of the speed is measured correctly $\alpha = 1$. If the speed is 10 % is to high, $\alpha = 1.1$ and so on.

Drag and Rolling Resistance Coefficient

In section 2.3.3 is the computation of the important drag and rolling resistance coefficient described.

What influence has α on the polynomial approach. lets rewrite the matrices with α .

$$\mathbf{A} = \begin{bmatrix} \frac{\rho}{2m} \alpha v_0^2 & g \\ \vdots & \vdots \\ \frac{\rho}{2m} \alpha v_n^2 & g \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -\alpha \hat{a}_0 \\ \vdots \\ -\alpha \hat{a}_n \end{bmatrix}, \mathbf{x} = \begin{bmatrix} C_D A \\ C_R \end{bmatrix} \quad (3.11)$$

The Factor $\frac{\rho}{2m}$ can be written as ξ . The calculation can be done for two datapoints step by step,

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} \alpha^4 \xi^2 (v_0^4 + v_1^4) & \alpha^2 g \xi (v_0^2 + v_1^2) \\ \alpha^2 g \xi (v_0^2 + v_1^2) & 2g^2 \end{bmatrix} \quad (3.12)$$

$$(\mathbf{A}^T \mathbf{A})^{-1} = \begin{bmatrix} \frac{2}{\alpha^4 \xi^2 (v_0^2 - v_1^2)^2} & -\frac{v_0^2 + v_1^2}{\alpha^2 g \xi (v_0^2 - v_1^2)^2} \\ -\frac{v_0^2 + v_1^2}{\alpha^2 g \xi (v_0^2 - v_1^2)^2} & \frac{v_0^4 + v_1^4}{g^2 (v_0^2 - v_1^2)^2} \end{bmatrix} \quad (3.13)$$

$$(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T = \begin{bmatrix} \frac{1}{\alpha^2 \xi (v_0^2 - v_1^2)} & -\frac{1}{\alpha^2 \xi (v_0^2 - v_1^2)} \\ -\frac{v_1^2}{g (v_0^2 - v_1^2)} & \frac{v_0^2}{g (v_0^2 - v_1^2)} \end{bmatrix} \quad (3.14)$$

$$(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T b = x = \begin{bmatrix} -\frac{\hat{a}_0 - \hat{a}_1}{\alpha \xi (v_0^2 - v_1^2)} \\ \frac{\alpha (\hat{a}_0 v_1^2 - \hat{a}_1 v_0^2)}{g (v_0^2 - v_1^2)} \end{bmatrix} \quad (3.15)$$

$$C_D A = -\frac{1}{\alpha} \frac{\hat{a}_0 - \hat{a}_1}{\xi (v_0^2 - v_1^2)} \quad (3.16)$$

$$C_R = \alpha \frac{(\hat{a}_0 v_1^2 - \hat{a}_1 v_0^2)}{g (v_0^2 - v_1^2)} \quad (3.17)$$

As you can see the factor α influences $C_D A$ as a inverse multiplication and C_R as a normal multiplication. The number of datapoints has no effect on the result.

Power

The power with its equation 2.7, can be rewritten with α ,

$$P = \left(m \alpha \hat{a}(t) + \frac{1}{2} \rho \frac{1}{\alpha} C_D A (\alpha v(t))^2 + \alpha C_R m g \right) \alpha v(t) \quad (3.18)$$

and that equals to

$$P = \left(m \alpha \hat{a}(t) + \frac{1}{2} \rho \frac{1}{\alpha} C_D A (\alpha v(t))^2 + \alpha C_R m g \right) \alpha v(t) \quad (3.19a)$$

$$P = \left(m \hat{a}(t) + \frac{1}{2} \rho C_D A v(t)^2 + C_R m g \right) \alpha^2 v(t) \quad (3.19b)$$

That concludes that the α has a quadratic influence on the power.

3.2.2 Numerical Approach

The Method to compute the power with a coasting and acceleration test run described in 2.3 has been implemented in python. The function can be used to show numerically how an error α affects the power of the car during the acceleration test.

In fig 3.9 can you see the measured velocity of the acceleration test. The error factor α is set to 1.1 thus an error rate of 10 %.

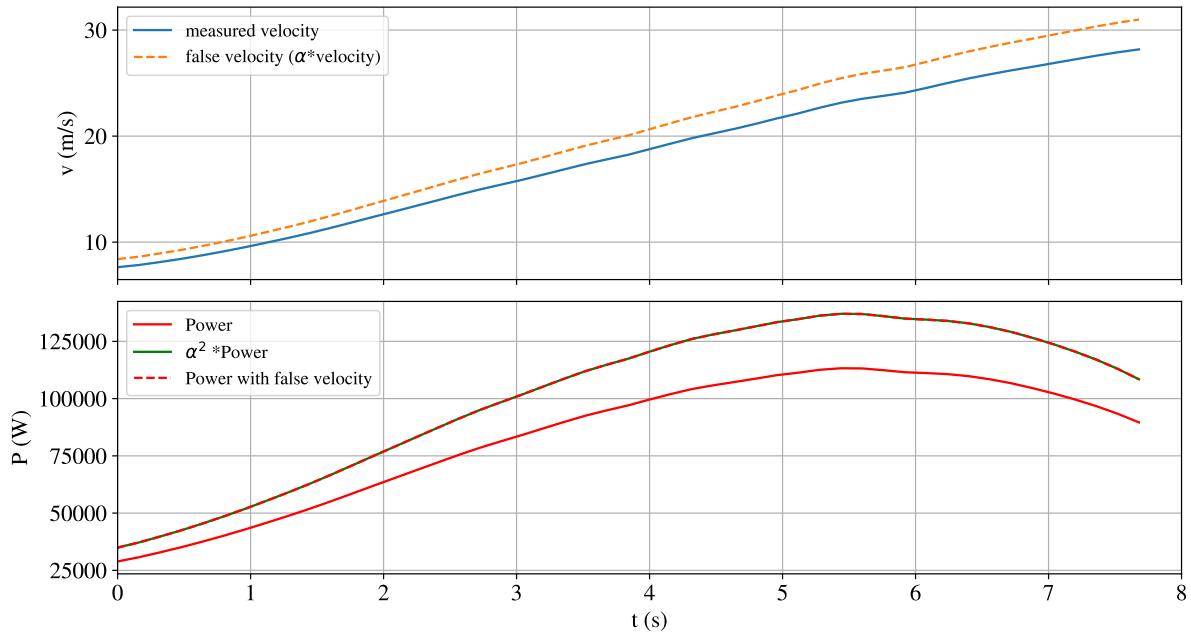


Figure 3.9: Power with falsely measured velocity

In the second subplot you can see the power and the power with the wrong velocity. In the previous section has be analytically shown that α has an quadratic effect on the power. To compare the analytical an numerical approach the normal power (red line) is multiplied with α^2 (green line). The green and dashed red line matches quite perfect. The maximal power is in 3.9 $\max(P) = 137030 \text{ W}$ and the maximal value of the wrong power is $\max(P_w) = 113250 \text{ W}$. The factor between the two maximal values

$$\alpha_{numerical} = \frac{137030 \text{ W}}{113250 \text{ W}} = 1.20997799 \quad (3.20)$$

and $\alpha^2 = 1.1^2 = 1.21$.

With this can be said, that both the analytical and numerical approach show an quadratic effect of a falsely measured velocity on the power.

3.3 Transmission Gear Ratio

All components connecting the motor to the wheels are part of the drivetrain. The purpose of the drivetrain is to transfer the generated power from the motor onto the ground. Next to drive shafts, differentials and other components, the gearbox represents an important part of it. [8] In

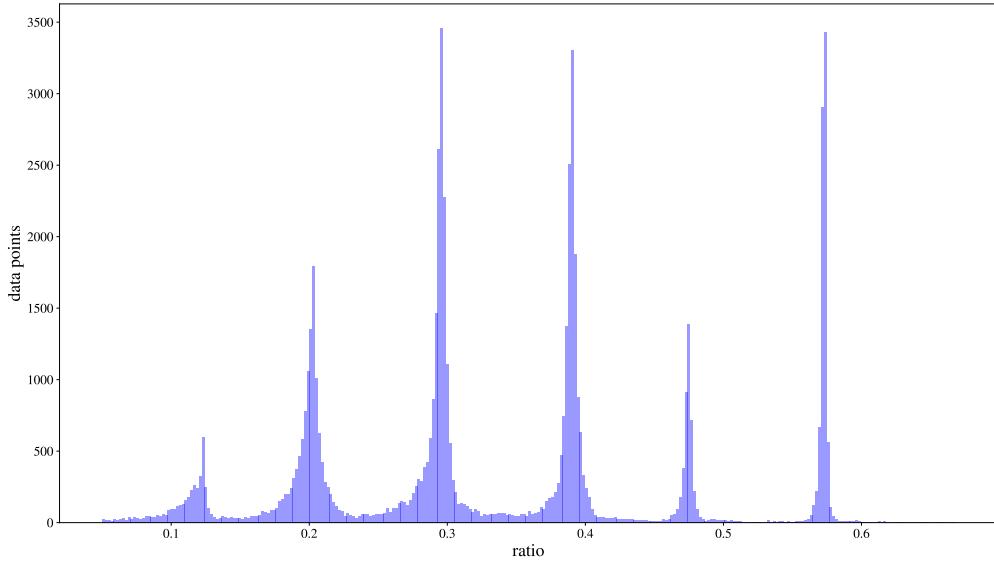


Figure 3.10: Histogram of the ratio from the rpm of the engine to the rpm of the wheels.

contrast to the differential, the gear box does not have a constant ratio. It depends on the used gear which by itself has a fixed transmission ratio each.

3.3.1 Delay by Histogram Matching

In order to get the ratio between the rotational speed of the motor and the wheel, the two signals has to be aligned. A similar approach as in 3.1.1 has been tested.

With the revolutions of the motor and the wheel, the motor-to-wheel-ratio can be calculated. In the histogram in figure 3.10 a number of significant peaks are visible. Each of these peaks represent a gear. Unfortunately, the neutral gear introduces unuseful data points because it separates the motor from the wheels. Hence, the vehicle speed does not correlate to the engines rpm anymore. For a power measurement with the Insoric Realpower module, a roll-out maneuver has to be done. The car needs to be accelerated to the maximum possible rpm and must then be put into neutral. During the roll-out phase, the motor drops down to its idle rotational speed. However, the vehicle rolls out from a relatively high velocity down to a full stop. This results in unuseful datapoints across the whole range, similar to a noise-band.

3.3.2 Gaussian Mixture Model

Visually these peaks are similar to a normal distribution, so the theory of a “mixture of gaussians” is introduced. A ‘mixture of gaussians’ is a collection of normally distributed probability densities with the common form

$$g(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right). \quad (3.21)$$

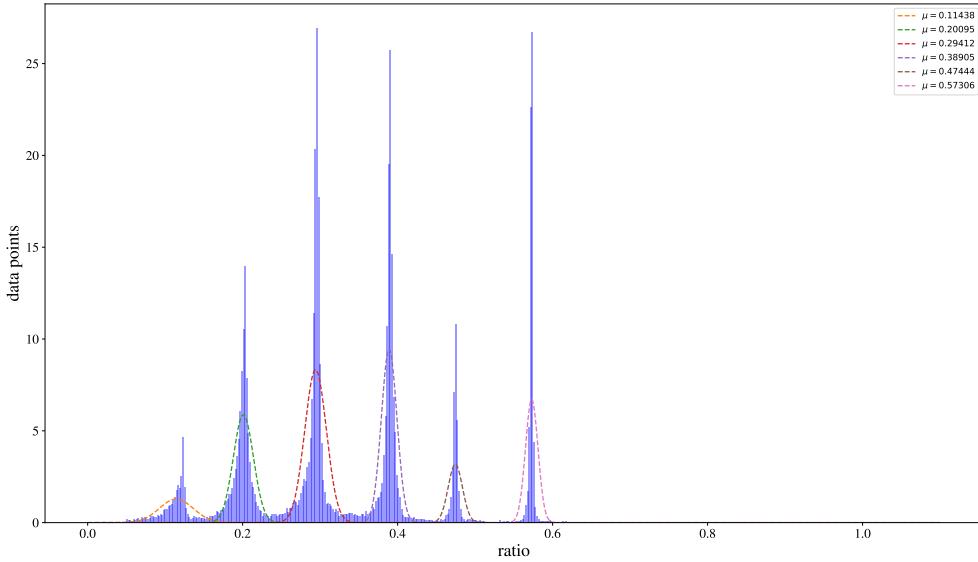


Figure 3.11: Application of the gaussian mixture model on the histogram

The GaussianMixture class from the python library ‘sklearn’ uses an expectation maximisation (EM) algorithm to find the best fit. The EM algorithm is an iterative method which finds maximum likelihood estimates.

Seemingly, this algorithm had difficulties detecting gears which haven’t been used long enough, thus not having much data points. For instance, the first gear mostly gets used to set the car into motion. Especially automatic gear transmission systems shift very fast into the second gear when accelerating in a normal fashion. Hence, only few data-samples are gathered, which leave the gaussian distribution quite uncertain. With only few samples it can hardly be separated from the noise-data. Because of this, the proposal was made in chapter 4, to make an initial drive before a power measurement in order to gather enough samples.

The gaussian mixture model fits different numbers of gaussian distributions on the given data using the expectation-maximization algorithm.

To evaluate the best fit, the bayesian information criterion (BIC) is introduced. The BIC is formally defined as

$$BIC = \ln(n) k - 2 \ln(\theta) \quad (3.22)$$

where n is the number of data points, k the number of parameters and θ the maximized value of the likelihood function of the estimated model. To prevent overfitting, the Bayesian information criterion puts a penalty on a high number of parameters of the model. Without this penalty, the model with a high number of components would always win, because the noise would be fit into the model as well.

The ratios of the gearbox are assumed to have a regular distance to each other in order to be spread evenly across the range of ratios. Thus, models with gears which are nearer to each other than an estimated threshold get punished harder than those having an expected distribution. So

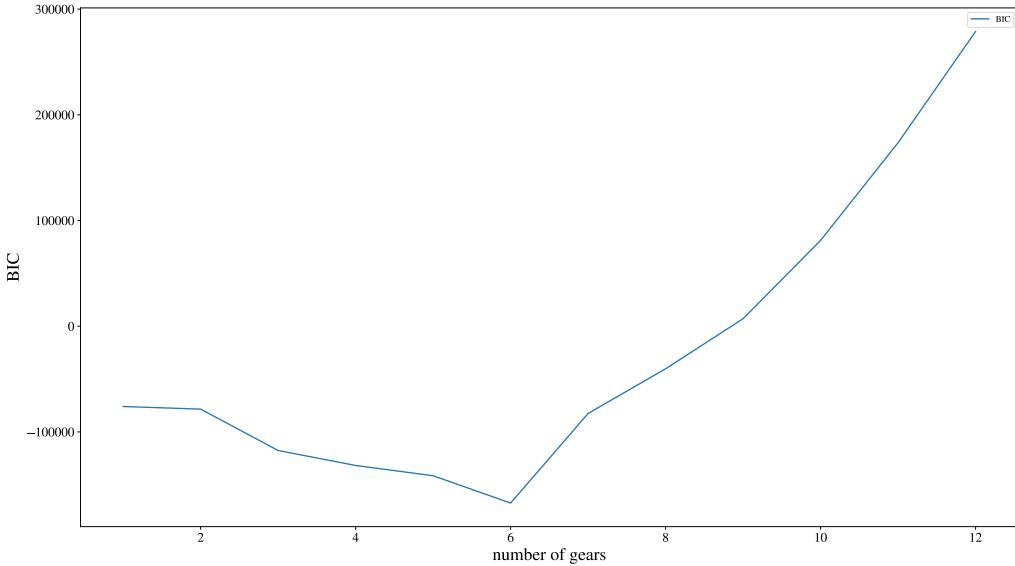


Figure 3.12: Visualisation of the slightly altered bayesian information criterion.

a exponential parameter p was brought into equation 3.22 resulting in

$$BIC = \ln(n) k^p - 2 \ln(\theta). \quad (3.23)$$

where p depends on the spacing between the mean values. Under this assumption the model performs more reliable than before. Note that this is not part of the true bayesian information criterion. In figure 3.12 the BIC was calculated for the fit of the ratio data on the GMM with various numbers of possible gears. The model with the lowest BIC gets picked out. In figure 3.11 the best fit would be the model with 6 components.

As you can see in figure 3.11, the significant peaks are detected. Visually the mean values do not always seem to be exactly where they should be. Furthermore, the standard deviation of the gaussians seem to be quite large. This is assumed to be because of the noise which automatically gets fitted to the model in order to minimize the BIC. In order to dispose of this, the ratios around the peak are thresholded so we get a more accurate result. Therefore, the gaussian mixture model gets used, to estimate how many gears have been used and roughly guess gear ratios. In a next step, the ratios around this estimated peaks get used to calculate a single gaussian distribution (3.21) in order to find a more exact peak, disregarding the noise.

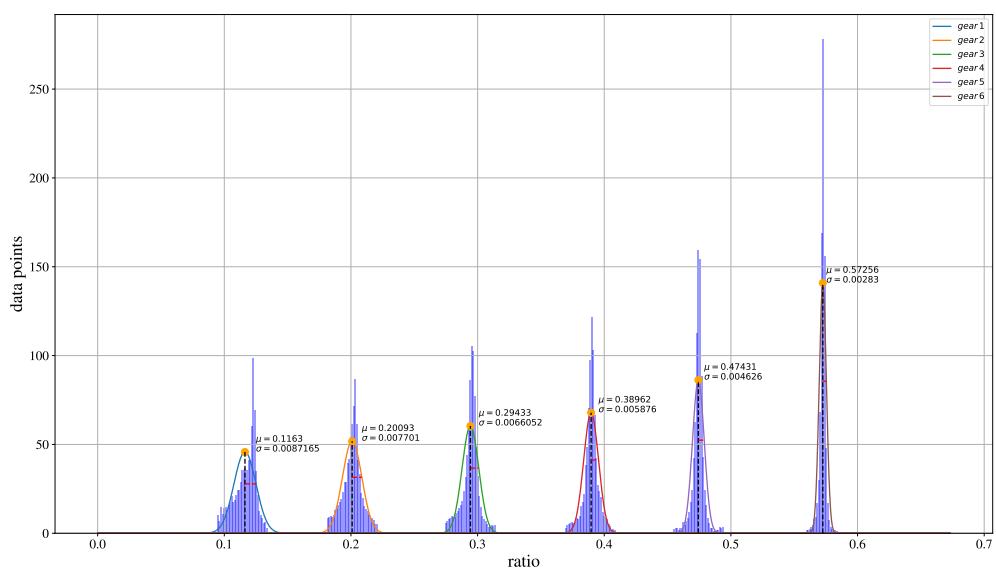


Figure 3.13: Application of the gaussian mixture model of the filtered histogram

Chapter 4

Recommendations

4.1 Initial drive

In order to gather enough data, an initial drive shall be done before a power measurement. For this initial drive, the operator of the vehicle is asked to drive for a few hundred meters in each gear, before shifting to the next one. During this drive he/she should keep a reasonable and constant speed for each gear. The velocity, of course, must not exceed the given speed limit or make the operator feel unsafe or put anyone in danger. With this initialisation, enough data-samples are gathered for the software to compute all the desirable information.

Recommendations

Chapter 5

Conclusion

Conclusion

Bibliography

- [1] “Newton second law of motion.” https://en.wikipedia.org/wiki/Newton%27s_laws_of_motion, november 2019.
- [2] M. Mitschke, *Dynamik der Kraftfahrzeuge*, vol. 4. Berlin Heidelberg: Springer, 2004.
- [3] “Rollversuche.” <https://mini.fandom.com/de/wiki/Fahrrad/Technik/Rollversuche>, november 2019.
- [4] “On-board diagnostics.” <https://www.obd-2.de/obd-2-allgemeine-infos.html>, november 2019.
- [5] “Obd pids.” <https://www.csselectronics.com/screen/page/simple-intro-obd2-explained/language/en#OBD2-PIDs-Intro-Data-Logging>, november 2019.
- [6] “Realpower module picture.” https://www.insoric.com/bilder/High_Res_Key_Visuals_jpg/Module-2.jpg, december 2019.
- [7] S. J. Orfanidis, *Introduction to Signal Processing*, vol. 1. New Jersey 07458 Upper Saddle River: Prentice Hall, 1996.
- [8] “Transmission_(mechanics).” [https://en.wikipedia.org/wiki/Transmission_\(mechanics\)](https://en.wikipedia.org/wiki/Transmission_(mechanics)), december 2019.

Conclusion

List of Figures

2.1	The check engine light (also called malfunction indicator lamp (MIL)) indicates a malfunction of the vehicle. https://commons.wikimedia.org/wiki/File:Motorkontrollleuchte.svg	
2.2	An OBD message is structured in a 75-bit-long bitstream containing a 11-bit-long Identifier followed by the data (64 bit). [5]	8
2.3	The RealPower Module [6]	10
2.4	Differential equation fitted to a coasting test run	12
3.1	Correlation	14
3.2	Failed correlation	15
3.3	Delay with MSE	16
3.4	Circumference sorted by speed	17
3.5	Circumference with averaging and linear regression	18
3.6	Circumference measurement with different wheel pressure	19
3.7	Probability density function of the circumference	19
3.8	Circumference computed with a different minimum threshold for the velocity	20
3.9	Power with falsely measured velocity	22
3.10	Histogram of the ratio from the rpm of the engine to the rpm of the wheels.	23
3.11	Application of the gaussian mixture model on the histogram	24
3.12	Visualisation of the slightly altered bayesian information criterion.	25
3.13	Application of the gaussian mixture model of the filtered histogram	26

Conclusion

List of Tables