Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2017

Real-Time Calibration of the Steering Wheel Angle Sensor

Nils Larsén

LINKÖPING UNIVERSITY

Master of Science Thesis in Electrical Engineering

Real-Time Calibration of the Steering Wheel Angle Sensor

Nils Larsén

LiTH-ISY-EX--17/5066--SE

Supervisor: **Olov Holmer**
ISY, Linköping University
Jacob Johnsson
Scania CV

Examiner: **Erik Frisk**
ISY, Linköping University

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

A stationary or temporary offset in the steering system of a vehicle can result in functions, relying on the steering wheel angle, performing poorly. Due to the wide range of different vehicle configurations at Scania CV, all sensors with relevant information regarding vehicle direction are not available on all vehicles. By using a statistical approach, including common sensors installed on the vehicle, a conceptual algorithm calibrating the Steering Wheel Angle Sensor offset in real-time has been developed. The algorithm is simple and relies on the assumption that a vehicle is driving straight ahead most of the time above a certain minimum vehicle speed, thus the most frequent steering wheel angle is the straight ahead angle. The algorithm is only active above the certain minimum vehicle speed and consists of two moving windows comprising steering wheel angle samples in which the calculations are performed. The results show that the algorithm is able to detect offsets with a short calibration time. Storage of samples is required but no vehicle specific parameters are needed.

Acknowledgments

Writing this thesis would never have been possible without my supervisor at Scania, Jacob Johnsson. I would like to express my sincerest gratitude to you for all the time that you have spent discussing the problem and helping me move forward. I would also like to express my gratitude to my supervisor at Linköping University, Olov Holmer and my examiner Erik Frisk who have been supportive and helped me with the report. My appreciations also go to the people at Scania REV in general and Scania REVT in particular for letting me write this thesis.

Södertälje, June 2017
Nils Larsén

Contents

Notation **ix**

1 Introduction **1**
1.1 Background 1
1.2 Problem Formulation 1
1.3 Objective 2
1.4 Delimitations 2
1.5 Previous Work 2
1.6 Outline of the report 3

2 Theoretical Background **5**
2.1 Steering Wheel Angle 5
2.2 Steering Mechanism 5
2.3 Tag Axle Steering 6
2.4 Sensors 7
2.5 Steering Wheel Angle Sensor Offset 8
2.6 Driving Conditions and Vehicle Dynamics 8
2.7 Segmenation 10
2.8 Statistical Analysis on Signals 10
2.8.1 Statistical Features 10
2.8.2 Running Statistics 11
2.8.3 Moving Windows 11
2.8.4 Weighted Windows 11

3 Data Analysis **13**
3.1 Data 13
3.2 Statistical Assumption 14
3.3 Straight Ahead Validation 14
3.4 Problematic Driving Conditions 16
3.5 Segmentation 16
3.6 Statistical Approach 18

4 Algorithm Development **21**

4.1 Algorithm 21
4.2 Algorithm Description 22
4.2.1 Slow Calibration 23
4.2.2 Quick Calibration 23
4.2.3 Offset Output 23
4.3 Parametrisation 24
4.3.1 Resolution 25
4.3.2 Weighting Parameters 25
4.3.3 Boundary Parameters 26
4.3.4 Window Size Parameters 26

5 Results **29**
5.1 Tests 29
5.1.1 Test A - Steady State 30
5.1.2 Test B - Small Step 31
5.1.3 Test C - Middle Sized Step 32
5.1.4 Test D - Large Step 33
5.2 Detailed Results 34
5.3 Calibration Time 35

6 Summary **37**
6.1 Discussion 37
6.1.1 Performance 37
6.1.2 Data 38
6.1.3 Memory Allocation 38
6.2 Conclusion 39
6.3 Future Work 40

Bibliography **41**

Notation

Nomenclature

Notation	Description	Unit
δ	Steering Wheel Angle	$^\circ$
δ_o	Steering Wheel Angle offset	$^\circ$
δ_s	Steering Wheel Angle slow offset	$^\circ$
δ_q	Steering Wheel Angle quick offset	$^\circ$
r	Yaw Rate	$^\circ/s$
a_1	Weighting Parameter	-
a_2	Weighting Parameter	-
a_3	Weighting Parameter	-
w_{slow}	Slow Window Size	-
w_{quick}	Quick Window Size	-
B_{low}	Low Boundary Parameter	$^\circ$
B_{high}	High Boundary Parameter	$^\circ$

Abbreviations

Abbreviation	Description
SAS	Steering Wheel Angle Sensor
SWA	Steering Wheel Angle
YRS	Yaw Rate Sensor
TAS	Tag Axle Steering
ESP	Electronic Stability Program
CAN	Controller Area Network
ECU	Electronic Control Unit

1 Introduction

1.1 Background

The Steering Wheel Angle Sensor, SAS, is an essential part in vehicles and is used to measure the Steering Wheel Angle, SWA. The information of the current SWA is needed by several functions in the vehicle such as the Tag Axle Steering, TAS, and the Electronic Stability Program, ESP, since they rely on the SWA. Along with time and use, offset errors might be introduced in the steering system affecting the SAS to supply an uncalibrated SWA. These offsets may occur temporarily during loading and unloading of the carriage or permanently as a consequence of tearing, service and deformation of the steering front axle.

Scania CV wants an investigation in a real-time offset estimation of the SAS that can be conformed to all trucks in their product range. Hence not all sensors with the direction related information can be used since some of them are only available on high specification trucks. The low specification requirement limits the problem to only use a few sensors such as the SAS as it exists on almost every vehicle.

1.2 Problem Formulation

The problem is developing a conceptual algorithm to detect and determine offsets in the SAS. The algorithm should be developed with a statistical approach in order to reduce the amount of sensors needed and should only be active during certain conditions. Since it is of interest to one day implement the algorithm on an Electronic Control Unit, ECU, memory requirements exist resulting in the inability of storing all input samples, which are accounted for during the development. Furthermore, the offset should be detected and calculated regardless of it being stationary or temporary.

1.3 Objective

With an offset error that may be introduced permanently or temporarily in the SAS the objective of this project is to develop an algorithm that determines the offset error regardless of origin. Furthermore, the algorithm should only use signals from sensors that can be found in low specified trucks.

1.4 Delimitations

Since other functions in the vehicle benefit from a calibrated SAS it could be of interest to further study the possibilities and challenges in these functions that may arise from this new information. Since the objective of this thesis is to only determine the offset, the function owners will need to handle this new information. However, some background information is given for the reader to understand why a calibrated SAS is of interest.

1.5 Previous Work

Previous work in the subject comprises mainly of patents. These patents all strive to estimate and calibrate the offset error of the SAS and use additional sensors and many vehicle specific parameters. In [1] the authors have developed a histogram-based method to determine the offset in the SAS. In short, samples from the SAS are collected and a first frequency distribution is calculated in predetermined intervals, then the average of the first frequency distribution is determined. The first frequency distribution is then checked against the average and if the values in the frequency distribution are distributed symmetrically around the average all the values from the first frequency distribution are moved to a second frequency distribution. If not, the values are not accountable for and a new first frequency distribution is calculated. As soon as the number of values in the second frequency distribution exceeds a certain threshold the SWA offset is calculated as the average of the second frequency distribution. This patent only accounts for values from the SAS when the vehicle is driving nearly straight ahead and uses additional sensors measuring lateral acceleration, yaw rate and vehicle speed to segment the signal.

In the patent [8] the authors have developed a method that adapts the SWA offset to changing road conditions. By taking gravity into consideration the method can identify banking, hence is able to correct the offset such that the driver does not need to compensate for the impact from gravity. In order to identify changing road conditions the method needs a plurality of sensors, measuring lateral acceleration, wheel speed and yaw rate. By modelling the SWA using different input signals, different modelled SWA outputs are obtained. After low-pass filtration the signals are weighted and the SAS offset is calculated.

Using additional sensors to model the SWA is also done by Peter Fejes [3] in his master's thesis. Briefly, he models the SWA using Ackermann steering geometry equations and compares it to the measured SWA to obtain an offset. Since the Ackermann steering geometry is inaccurate under certain conditions, a set of requirements need to be fulfilled for his algorithm to be active such as minimum speed and minimum yaw rate.

1.6 Outline of the report

In Chapter 2 research and relevant information needed for the project is presented. This chapter aims to give the reader an understanding on the fundamentals of the system and prerequisites in order to facilitate continuous reading.

In Chapter 3 the data used for the algorithm is shown and the statistical assumption, on which the algorithm is based on, is presented. The driving scenarios that will influence the algorithm negatively are described and a segmentation approach, to handle some of the problematic driving scenarios, is visualised. Also, some prerequisites that will simplify the understanding of the algorithm are discussed.

The developed algorithm is presented in Chapter 4 as well as its performance and validation.

The final results are presented in Chapter 5 and discussion and future work can be read in Chapter 6.

2

Theoretical Background

In this chapter the theoretical background needed for project is presented. Relevant techniques needed to process the problem are all described here.

2.1 Steering Wheel Angle

When a truck leaves the assembly line, calibration of the SAS is done by aligning the front wheels with the truck chassis and adjusting the steering wheel. The SWA is the angle measured by the SAS and by multiplying this angle with a steering ratio the angle of the front wheels can be obtained. Thus having a steering ratio of i.e. 20:1 would cause the front wheels to turn 18 degrees if a full turn (360 degrees) on the steering wheel is completed and no play in the steering system exists. The SAS is essential for several functions in the vehicle as it holds information on the direction of the vehicle. The TAS uses this information to steer the tag axle, enabling a shorter turning radius and less tear on the tires.

2.2 Steering Mechanism

The mechanical components that transmit torque from the steering wheel to the front wheels constitutes the steering system, see Figure 2.1. Briefly, the steering wheel is mounted on the steering column that enables torque transmission to a steering gear [5]. The steering gear amplifies the torque and steers the front wheels at a ratio of approximately 20:1 in common heavy duty vehicles [11]. Since the components are attached by joints and the steering gear is a hydraulic system some play exists in the steering system [3]. The potential play arises due to non-linearities and needs to be considered since it implies that applying a certain torque on the steering wheel may not be manifested on the front wheels even if the SAS indicates steering [6].

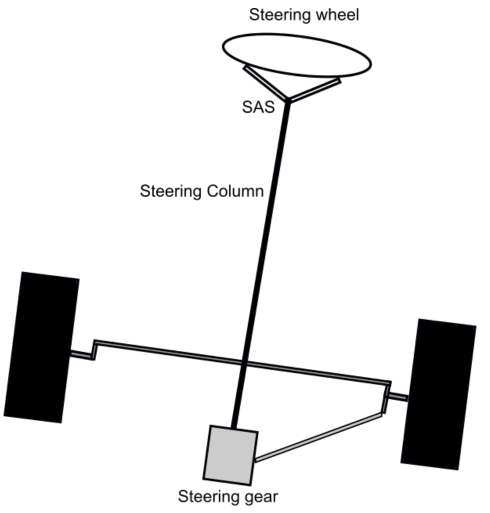

Figure 2.1: *A simplified sketch of a typical steering system.*

2.3 Tag Axle Steering

This project is not about the Tag Axle Steering, TAS, but since the TAS relies on a calibrated SAS a brief description of how it works and why it needs the SWA to function properly is presented.

The TAS is a function, mainly in trucks, that enables steering of the wheels at the tag axle. The steerable tag axle decreases the turning radius which enhances driving ability for the driver and less tearing on the tires as the rate of slip is reduced.

The TAS is not active in an interval of -X°$< \delta <$ X° due to uncertainties of the actual SWA. If an accurate offset estimation of the SWA can be achieved, this window can be reduced and the TAS function may work better since for higher longitudinal vehicle speeds, the turning of the steering wheel tends to be smaller. This means that cornering in higher speed may not activate the TAS since the rotation will be in the uncertainty interval. Furthermore, an offset in the SAS does not lead to an offset of the uncertainty interval since it is symmetric around 0°thus the offset will result in a skewed TAS uncertainty interval. In Figure 2.2 the principle is shown. The interval to the left and in the middle is the behaviour of the uncertainty interval for an uncalibrated SAS. With an offset in the SAS it would require less rotation of the steering wheel in one direction to activate the TAS while rotating the steering wheel in the opposite direction would require a larger turning to activate the TAS. The interval to the right is how this interval can be handled when information about the SAS offset is present. If the offset is zero, the interval is symmetric around zero, else symmetric around the offset.

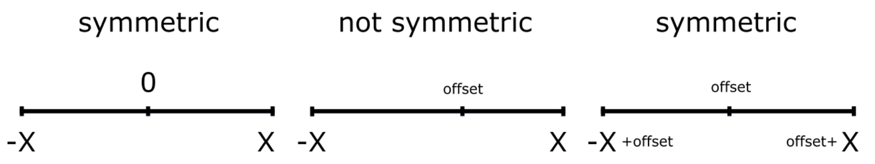

Figure 2.2: *Left is when no offset occurs. In the middle an offset occurs but no calibration is made. To the right an offset occurs and calibration is made and the interval is symmetric.*

2.4 Sensors

The sensors of interest for the algorithm are the Steering Wheel Angle Sensor, SAS and the longitudinal speed sensor. Both sensors are available on all Scania vehicles. The SAS is mounted just behind the steering wheel on top of the steering column and is a rotating sensor measuring the absolute steering angle. The front wheel angle is (simplified) determined by the previously mentioned steering ratio. A third sensor, Yaw Rate Sensor, YRS is also of interest in this report. Since the developed algorithm needs validation, the YRS can be used as it measures the movement around the Z-axis thus indicating the headed direction of the vehicle.

All signals are transmitted in the vehicle via CAN to ECU where the signals are used as input to various functions, such as the previously mentioned TAS.

In Table 2.1 the properties of standard sensors are provided. It should be noted that the resolution of the SAS has a higher resolution than the requirements for the algorithm output.

Table 2.1: *Sensor properties of common sensors.*

Sensor	Resolution	Range
SAS	0.1°	1500°
YRS	0.1°/s	100°/s

In Figure 2.3 the directions of a truck is visualised. The longitudinal vehicle speed sensor is measuring the speed along the X-axis. The SAS and the YRS is measuring the rotation around the Z-axis. Along the Y-axis, lateral forces are acting, such as crosswinds and centrifugal forces.

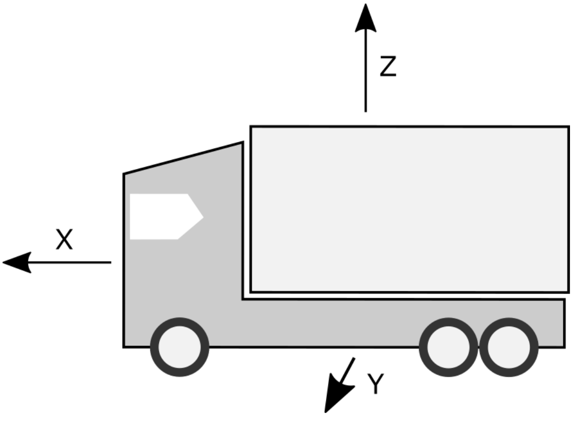

Figure 2.3: *Directions of a truck. Longitudinal forces on the X-axis, lateral forces on the Y-axis and vertical forces on the Z-axis.*

2.5 Steering Wheel Angle Sensor Offset

The SAS offset is defined as the deviation, in degrees, from the zero position when the vehicle is travelling straight-ahead. The SAS offset can appear in two ways: stationary and temporary.

The stationary offset typically occurs as a consequence of deformation of axis or stays in the chassis, tire wear or service. The stationary offset persists until service is done.

The temporary offset can occur during loading of a vehicle. The forces from the load affect the positions of the wheels and if the load is not evenly distributed it can effect the steering. Also certain driving conditions can be defined as temporary offsets which are presented in the next section.

One way of determining an offset in a sensor is by mathematically modelling the system. By then calculating the residual, i.e the deviation from the measured sensor signal with the modelled system an offset can be obtained. The accuracy of the offset depends partly on the accuracy of the measurement and the accuracy of the model. Noise or measurement errors in the sensor signals and model errors results in an inaccurate offset.

Mathematical models and theory of the steering of a vehicle is extensively covered in many books such as [12]. In this thesis the approach of modelling the steering is not of interest. Instead a statistical approach will be performed.

2.6 Driving Conditions and Vehicle Dynamics

Several driving conditions can be assumed problematic as they affect the steering of the vehicle. To begin with, the longitudinal vehicle speed affects the steering.

In speeds below 60 km/h the torque applied on the steering wheel results in cornering of the vehicle, and with reduced speed a higher torque can be applied on the steering wheel, resulting in ranging of the vehicle. Speeds above 60 km/h decrease the SWA [5] as forces affecting the vehicle disable the possibilities of applying certain torques on the steering wheel without risking rollover.

Crosswinds affect the steering such that the driver has to countersteer towards the wind direction. In order for the crosswinds to affect the steering and result in a new offset, the winds need to be persistent. Gusts that influence the steering for shorter periods of time will presumably not result in an offset. In theory, an understeered vehicle does not need to countersteer when the pressure point from the crosswinds is behind of the centre of mass of the vehicle [10]. In reality however, the pressure point from the crosswinds more often hit centre of mass thus countersteering is necessary in strong crosswinds.

Banking and cross slopes also have a certain impact on the steering. Both banking and cross slopes have the characteristic of an inclination perpendicular to the direction of the road. Cross slopes are used in order to drain water more efficiently and according to [9] a 2% to 2.5% slope angle is the most widely used design. Banking appears on curves to deal with centrifugal forces that act on the vehicle and are usually steeper than cross slopes. The inclination angle will have an effect on the lateral acceleration according to (2.1) where g is the gravitational constant and θ is the angle of the road inclination.

$$a_y = gsin(\theta) \tag{2.1}$$

Equation 2.1 states that a steeper inclination of the angle results in a higher lateral acceleration when the vehicle is driving straight-ahead and no other forces is acting on the vehicle. It should be noted that vehicles are configured to prevent being affected by cross slopes, but there are regions where the roads have a steeper inclination than the design policy [9], hence the driver will need to countersteer in order to drive straight and a SAS offset might be induced as a consequence. Vehicles with active suspension steering are often configured to compensate for crosswinds and cross slopes [5].

Ring-roads or long curves are also worth mentioning. Driving in such conditions might be interpreted as a temporary SAS offset corresponding to the most frequent angle of the steering wheel if the vehicle is turning with constant radius and speed for a longer period of time.

Finally toe needs to be mentioned as it might affect the directional stability of the vehicle [2] and thus the steering behaviour. Toe can occur as toe-in, i.e. the direction of the wheels rotates around the Z-axis inwards. Toe-out is when the wheels are rotated outwards. Often heavy duty trucks are configured to have a slight toe-in when unloaded. Research then suggests that the amount of toe-in decreases with increased axle load, e.g. loading or cornering [2].

Presumably there are other factors that may have an impact on the steering of the vehicle but they have not been studied in this thesis.

2.7 Segmenation

Segmentation is a pattern recognition technique that uses a data driven approach to classify unknown segments of a filtered signal that has been partitioned. When the feature extraction from the segments has been made, each segment is classified. Then segments that should be classified are partitioned and features are extracted from the segments. The segmentation can either be manual; all scenarios are predefined, or automatic; the scenarios are detected by adaptive thresholds that when exceeded create a new segment. Since a vehicle is in operation for several hours a day automatic segmentation is favourable since predefining all different scenarios is a time consuming work. The authors of [7] proposes a standard deviation technique to segment the signal over an interval. In their paper the adaptive threshold is also determined as the standard deviation of the observed signal in a predefined window size. The signal is then compared with the calculated threshold. If the signal is within the boundaries of the adaptive threshold the samples are included in an algorithm while samples exceeding the thresholds are not included.

The manual segmentation can be used as validation in simulation should the signals be known in advance [7]. The manual segmentation can also be used during development to facilitate the working process by predefining the thresholds. If the segmentation then is not correct the corresponding conditions can be segmented by automatic segmentation.

2.8 Statistical Analysis on Signals

To facilitate reading later in the thesis some prerequisites in statistics are presented in this section.

2.8.1 Statistical Features

A histogram visually represents the frequency distribution of a set of observed samples in predefined sized bins and is a convenient method to study a set of data before processing is made. By putting the collected samples in predefined bins, discretisation of the data is performed and a statistic - a value that summarises a set of values [4] - can be obtained from the set such as the mean or the MODE. By studying the histogram the most appropriate statistic can be chosen.

Measuring the average of a set of data using the mean approach is well motivated if no outliers exist and the samples are fairly normal distributed. However, a distribution with outliers will affect the average value in such a way that it may be misleading since the average will be shifted as the mean is sensitive towards outliers.

If instead the MODE of the distribution is used the average becomes a more trustworthy statistic of the distribution since the MODE is not as sensitive towards outliers. The MODE is the most frequent value, i.e. the value on the X-axis that has most number of observations on the Y-axis, and works properly in unimodal distributions, i.e, when only one peak occur in the data set. However, when the data is a multimodal distribution two MODE values are obtained which is undesirable as in this application only one value is wanted. For example, if a driver would drive in a sinus shaped road with the steering wheel positioned at -X° for left turns and X° for right turns a multimodal distribution is likely to occur in the SWA set.

2.8.2 Running Statistics

When calculating statistics in a real-time application previous samples need to be stored. If the mean is to be calculated then number of samples and the accumulated sum need to be stored in order to obtain the mean at each run. A similar approach when calculating the mode, i.e. the most frequent value in a window, can be done by storing the samples in an array and then calculate the mode each time the function is being run.

2.8.3 Moving Windows

As it could be of interest to calculate a statistic over a set of relatively fresh data a moving window can be used, see Figure 2.4. By predefining the length of the window the statistic will only be calculated in this window. This moving window then moves along the X-axis by removing the oldest sample in the set of data each time a new sample is added. By forgetting the history of the data new peaks in the set can be detected faster should they occur. A moving window also reduces the amount of stored samples.

2.8.4 Weighted Windows

Another way of dealing with the history of the data is to use a weighted window. By weighting the position of the samples in the window differently a desirable result can be achieved accordingly. This window can either be moving or not, and the weighting can be of fixed or exponential type, see Figure 2.5. An exponential moving window calculates the statistic by setting weights that change exponentially and never reach zero, equivalent to an infinite impulse filter. By doing so all history is included, with the newest samples having a bigger impact on the result. The weighting requires extra parameters that need to be predefined.

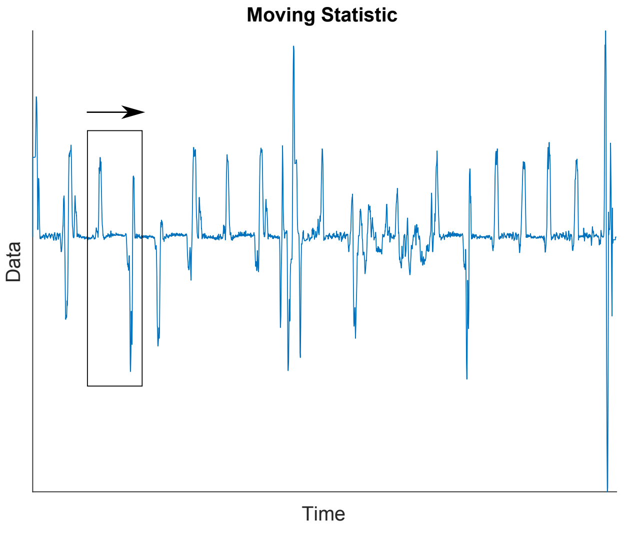

Figure 2.4: *Figure showing the principle of a moving statistic. As new samples are introduced, the window with fixed size moves along the X-axis and a statistic is calculated over the samples within the window.*

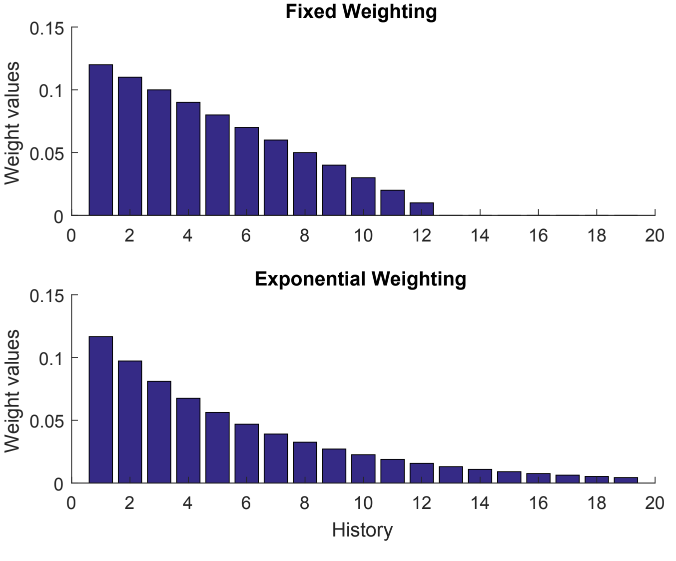

Figure 2.5: *The chart on top shows an example of fixed weighting parameters. If history is over 12 the samples are not included. The bottom chart shows an example of exponential weighting parameters. Note that these parameters never reaches zero.*

3 Data Analysis

This chapter aims to analyse and motivate the decisions and techniques used in the algorithm. First the data used in the report are presented. Then the statistical assumption that the algorithm relies on is presented. Driving conditions that may affect the steering, and thus the performance of the algorithm, are identified and a basic segmentation approach is described and visualised. Furthermore, the selected statistical approach is discussed to give an insight about the difficulties and compromises that need to be made.

3.1 Data

The data used in the report was recorded via CAN in a previous project at Scania and comprises SAS signals, longitudinal vehicle speed signals and YRS signals. The logged data is an approximate 35 minutes long log with and each signal has a sampling time of 20 ms.

In Figure 3.1 the measured data is visualised. As can be seen, when the longitudinal vehicle speed is low, the SWA tends to be higher than for higher speeds. This is expected since turning or ranging usually occur for lower speeds. Also depicted in the figure is the YRS which is used for validation. As the signal is rather noisy, filtering the signal is necessary. The SAS and longitudinal vehicle sensor are input signals for the algorithm and do not need filtering as the signals are sufficiently clean.

The data includes straight ahead driving, cornering and frequent steering at different longitudinal vehicle speeds.

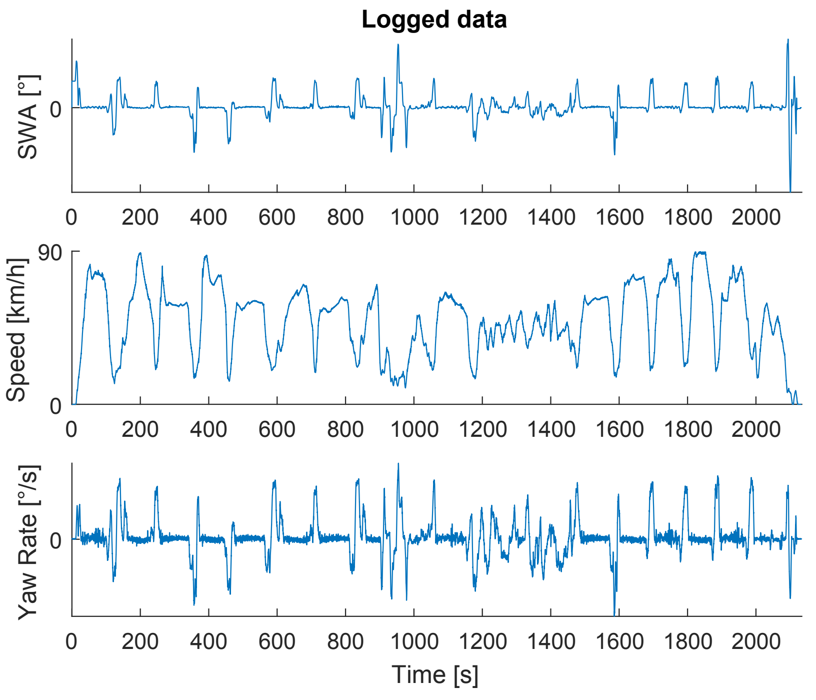

Figure 3.1: *The log containing SAS, speed and YRS signals. The grading of the Y-axis has been removed to minimise exposure of parameter values in the algorithm.*

3.2 Statistical Assumption

The statistical assumption on which the algorithm is based on is that a vehicle, when driving, is travelling straight ahead most of the time. Thus the most frequent angle of a set of SWA is the straight ahead angle. Should this value not be zero an offset has been introduced to the system.

Straight ahead direction is defined in this thesis as the SWA when the vehicle is travelling straight ahead. That is, it is not the actual position of the steering wheel that is of interest, nor the front wheels alignment with the chassis.

3.3 Straight Ahead Validation

By using the YRS signals from the data log, another source of information is obtained regarding the direction of the vehicle. With this redundant information, a conclusion whether the most common SWA is the straight ahead SWA or not can be drawn and the algorithm may, or may not, be based on the statistical assumption. As the YRS holds information of the vehicle direction it can be used to classify the SWA samples in either straight-ahead labels or not straight-ahead labels according to Table 3.1, where a strategy is presented to validate the following:

1. The most frequent angle is the straight ahead angle.
2. The SWA MODE is the correct offset.

The YRS signal is first low pass-filtered to reduce measurement noise. Due to the time-delay introduced by the filter, the SWA samples are delayed to match the corresponding YRS sample.

Table 3.1: *Strategy of validating the SAS offset with YRS.*

YRS	SAS	Offset
$r \approx 0°/s$	$\delta \not\approx °$	Yes
$r \not\approx 0°/s$	$\delta \approx 0°$	Yes
$r \approx 0°/s$	$\delta \approx 0°$	No

If δ is deviating from 0° and r is fairly constant around 0° indicates an offset in the SAS. Figure 3.2 validates that when $\delta \approx 0°$ the yaw rate is within its straight ahead boundaries, thus the calculated offset is the straight-ahead angle and in this set of data $\delta_o = 0°$. Given this information, the statistical assumption holds and can be used as a foundation to the algorithm.

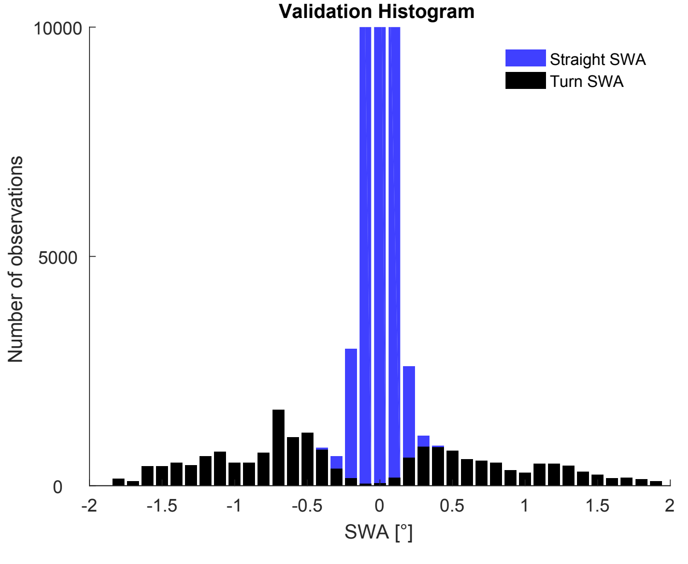

Figure 3.2: *The blue bars correspond to SWA samples where the YRS samples are within the straight-ahead boundaries. The black samples are the YRS samples exceeding the straight-ahead boundaries.*

3.4 Problematic Driving Conditions

To conclude section 2.6 the driving conditions that have an impact on the steering are:

1. Vehicle at standstill
2. Vehicle travelling at low speed
3. Crosswinds
4. Cross slopes
5. Ring roads

Condition 1 has been added to the problematic driving conditions since the statistical assumption is inaccurate if the vehicle is at standstill with the engine on.

In order to detect scenario 1-2 the longitudinal speed signal is needed since the SAS can not solely detect if the vehicle is travelling at a speed where ranging is most likely to occur or is at standstill. By a simple segmentation, conditions 1-2 can successfully be detected, which is further described in section 3.5. Conditions 3-5 however might be interpreted as temporary SAS offsets by the algorithm should they persist for a longer period of time as the driver has to countersteer which need to be considered during the algorithm development.

3.5 Segmentation

As previously stated the problematic conditions of when the vehicle is at standstill and when the vehicle is travelling at low speeds can be removed by segmenting the signal. Studying Figure 3.3 it is obvious that higher angles on the steering wheels occur for low speeds. In this data set, speeds above 30 km/h only includes SWA samples within approximately [-2,3] degrees. At speeds above 40 km/h this interval is reduced to approximately [-1.5,1.5]. It can also be seen that speeds above 60 km/h are subject to almost no steering input, as mentioned earlier [5].

The figure results in the conclusion that an appropriate minimum speed is in the range of 30 to 40 km/h in order to avoid samples with meaningless information. Although the studied data is only a 35 minutes long log it still gives valuable insight in the approximate behaviour of the data.

Here a basic segmentation is performed by using the information of the vehicle speed since the SAS signal can not solely detect conditions depending on speed. In Figure 3.4 it can be seen that the segmentation successfully separates the samples into grey areas (algorithm active) and white areas (algorithm inactive). By doing this segmentation, the algorithm does not store samples with misleading or unnecessary information concerning the SWA and a more reliable offset can be determined. As can be studied in the figure, the black line, corresponding to the SAS signal, SWA samples (blue line) that deviates from zero are removed except for samples between t $\approx$ 1200 to t $\approx$ 1500.

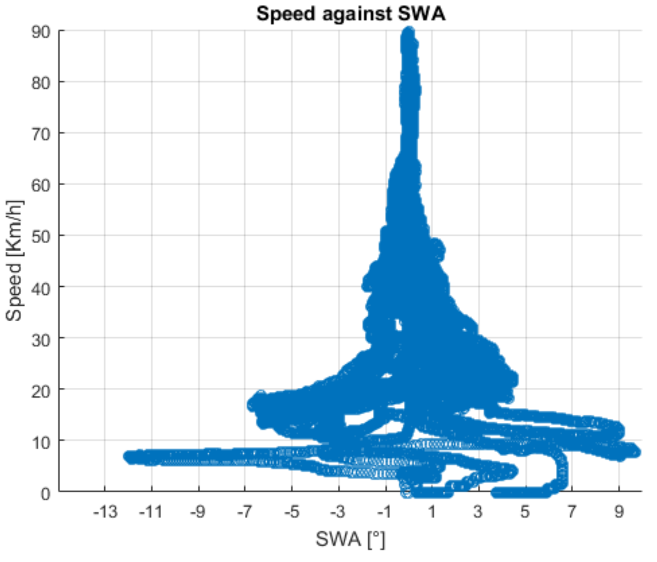

Figure 3.3: Speed plotted against SWA samples.

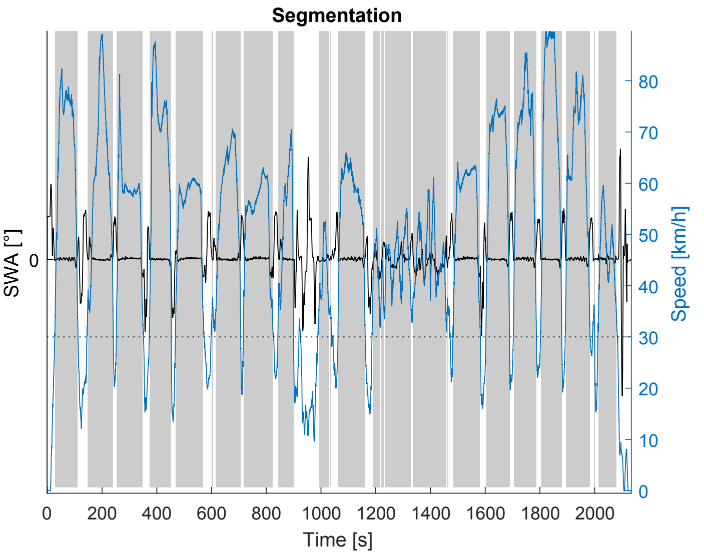

Figure 3.4: Example of the segmentation over a set of data. The algorithm is active at the grey areas and inactive at the white areas.

3.6 Statistical Approach

To find a value that represents the average of a set of data, the MODE or the mean can be calculated. Calculating an accurate mean requires that no outliers exist in the data set as they have a negative impact on the mean value, leading to a shifted average value. Since it is not certain that the segmentation will be able to remove all outliers in the SWA data, calculating the mean is not considered even if this is the fastest procedure concerning time complexity.

Regarding the alternative of using weighted windows it is not necessarily interesting to know the information of the SWA samples way back in history. To acquire a steady algorithm that still manages to detect new offsets only a short history of SWA samples is used and there is no need in weighting the samples in this short time frame. Using an exponential moving average could be a decent method when determining the offset. However, a trade-off is needed concerning steadiness (smoothness) and calibration time. In (3.1) a basic formula for an exponential moving average can be seen where EMA stands for exponential moving average, W is the smoothing factor and $\delta(t)$ is the current SWA sample.

$$\mathrm{EMA}(t) = W * \delta(t) + (1 - W) * \mathrm{EMA}(t-1) \tag{3.1}$$

Tuning of the smoothing factor W is needed and with a low W value a steady offset may be achieved but with a long calibration time. A high W value results in an oscillating behaviour but with quick responsiveness if an offset should occur. This is highly undesirable since the oscillating behaviour will continue at the new offset. Had the vehicle been travelling perfectly straight ahead this method could have been useful since storing samples is not needed and the smoothing average would be an accurate interpretation of the offset. However, the vehicle is not travelling perfectly straight ahead and it is still the most frequent angle that is to be found where the exponential moving average smoothens the data into an average to find a trend, rather than determining an exact offset.

Calculating the median as statistic is also a possibility, but since the samples are not sorted in the algorithm it is not preferable since a sorting procedure would increase the time complexity.

Calculating the MODE is not affected by outliers and is therefore a reasonable statistic. The MODE operation requires storage of data samples, compared to the exponential moving average, and a hardware implementation would result in a fairly harsh storage limitation. As this algorithm is developed in MATLAB no such requirement exists. However, as it is of interest to implement the algorithm in a real vehicle this storage requirement is still important to have in mind during the development to facilitate future implementation. Therefore, given the stored samples, a frequency distribution can be constructed in which the MODE can be calculated. The storage requirement further makes it impossible to store all samples, thus only an arbitrary amount of samples can be stored, hence a moving window is can be used which also enables a certain responsiveness if a new offset should occur. Since the MODE is the mathematical interpretation of the stated statistical assumption it is considered as the most appropriate statistic to use for the algorithm.

The segmentation performed earlier results in the histogram to the right in Figure 3.5. The outliers have successfully been removed in the studied data set compared to the left histogram which depicts the distribution before segmentation. The segmentation will therefore result in a quicker calibration as unnecessary samples are not stored.

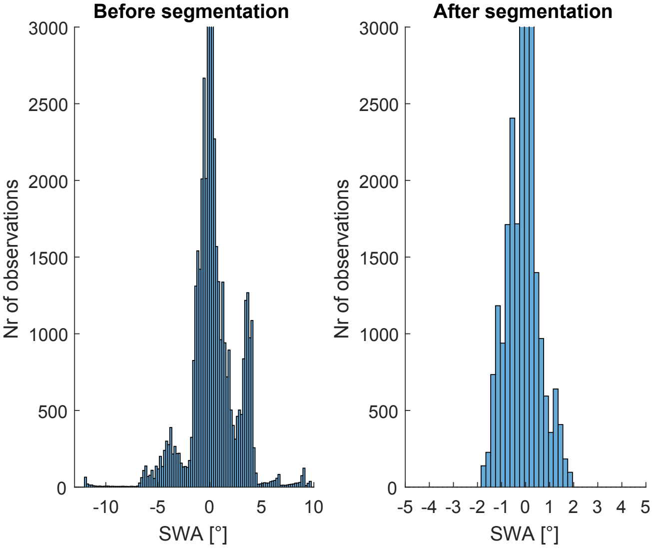

Figure 3.5: *Histograms showing the original data to the left and the segmented data to the right. The histograms are zoomed in at the base of the distribution.*

4

Algorithm Development

In this chapter the developed algorithm is presented. The prerequisites from Chapter 2 and the identified conditions and requirements from Chapter 3 is the foundation of the algorithm.

4.1 Algorithm

In Figure 4.1 the developed algorithm is visualised in a flow chart. The different parts of the algorithm are described in the following sections but a brief description is presented here.

The input signals from vehicle sensors are the SAS signal and the longitudinal vehicle speed signal. No filtering is performed on the input signals. However, in the MODE Calculation box, the binning of the SWA samples are made into a predefined resolution. The Slow MODE and the Quick MODE need predefined window sizes w_{slow} and w_{quick} where $w_{slow} > w_{quick}$. The Quick MODE needs a significant peak to obtain a new value, otherwise the old value is used. When values from the slow and the quick part have been calculated the absolute difference between the values are calculated. Depending on that difference, the contributions from the slow and quick part are weighted differently in order to achieve a quick calibration if such is needed or a steady calibration if such is needed. The final offset is then calculated by weighting itself, the slow offset and the quick offset dependent on the absolute difference between the slow and quick offset.

The algorithm has no vehicle dependent parameters but instead algorithm specific parameters that need to be predefined, see Section 4.3.

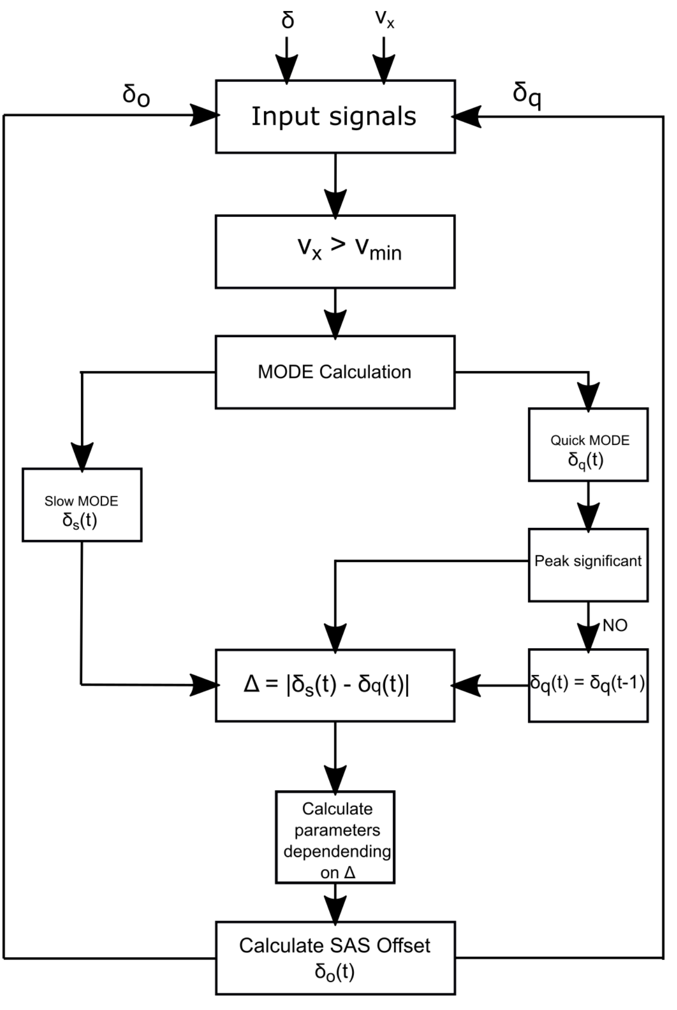

Figure 4.1: *A flow chart visualising a simplified version of the developed algorithm.*

4.2 Algorithm Description

As previously stated the aim of the algorithm is to find the most frequent SWA value and then calculate the current offset in the SAS sensor. To achieve this, the algorithm consists of two moving windows; one responsible for the slow calibration part calculating δ_s, containing more SWA samples; another one with fewer SWA samples responsible for the quick calibration part calculating δ_q. Both δ_s and δ_q are the MODE in respective window. By then combining the results of the current MODE from the windows the algorithm effectively manages to detect the offset, while at the same time manages to stay steady.

4.2.1 Slow Calibration

The purpose of the slow calibration part of the algorithm is to ensure a stable offset, insusceptible to external disturbances. With more samples in the window the probability is higher that the straight-ahead scenario is detected, and thus the offset. Another advantage with more samples in the window is that the MODE will not be subject to quick changes, thus a more steady output from the slow calibration can be achieved. The slow calibration part gives

$$\delta_s(t) = \text{MODE}\Big(\delta(t), \delta(t-1), \ldots, \delta(t - w_{slow})\Big) \tag{4.1}$$

where t denotes the time-stamp for the current sample and w_{slow} is a predefined parameter.

4.2.2 Quick Calibration

The purpose of the quick calibration part of the algorithm is to ensure quicker detection of a new offset in the system. Since the window, in which the MODE is calculated, is smaller than the slow window, the MODE is more likely to change its MODE quicker than the slow window if an offset appears. This is simply due to the fact that if a new offset appears, the quick calibration part will only consist of samples with shorter history comparing with the slow calibration part that will consist of samples with older history as well.

Since the quick calibration is rather transient due to the small window the characteristics of the steady slow calibration part is needed. Also, due to the volatility of the quick calibration part, a certain requirement must be fulfilled. This requirement states that the peak of the found MODE must be significantly higher than the peak of the second MODE. This requirement is necessary in order to avoid uncertainties. For example, if the vehicle is cornering, and all samples belonging to the turn are being used to calculate the MODE no significant peak will be found which is desired as the vehicle is not travelling straight-ahead.

The quick calibration part gives

$$\delta_q(t) = \begin{cases} \text{MODE}\Big(\delta(t), \delta(t-1), \ldots, \delta(t - w_{quick})\Big), & \text{if peak is significant} \\ \delta_q(t-1), & \text{otherwise} \end{cases} \tag{4.2}$$

where t denotes the time-stamp for the current sample and w_{quick} is a predefined parameter. The significant peak condition is formulated to be when the peak of the MODE is at least twice as high as the second highest peak.

4.2.3 Offset Output

In order for the algorithm to weight the contributions from the slow and quick calibration parts, a couple of parameters need to be determined, see Table 4.1. In

(4.3), Δ is calculated as the absolute difference between δ_s and δ_q and normalised with a boundary value parameter B_{high}, see Section 4.3 for algorithm specific parameters.

$$\Delta = \frac{|\delta_s(t) - \delta_q(t)|}{B_{high}} \tag{4.3}$$

The Δ is used to determine the weighting parameters a_2 and a_3 during run-time. The weighting parameters are used for the final output of the algorithm in order to determine the contributions from the slow and quick part. In (4.4) the strategy to determine a_2 and a_3 can be observed. If the first statement is true then a_3 is set to zero, i.e. the contribution from δ_q is none. If the second statement is true then the weighting sets δ_s to zero. Otherwise no weighting parameter is set to zero, resulting in contributions from both the slow and quick part.

$$a_2, a_3 = \begin{cases} (1 - a_1), 0, & \text{if } \Delta < B_{low}/B_{high} \\ 0, (1 - a_1) & \text{if } \Delta > 1 \\ (1 - \Delta)(1 - a_1),\ \Delta(1 - a_1) & \text{otherwise} \end{cases} \tag{4.4}$$

To summarise, a_2 and a_3 are determined by the conditions in (4.4) and together with the predefined parameter a_1 the following is true

$$\sum_{i=1}^{3} a_i = 1 \tag{4.5}$$

With the parameters determined and the offset contributions calculated from the slow part, δ_s and the quick part, δ_q, the offset output is calculated according to (4.6).

$$\delta_o(t) = a_1 \delta_o(t-1) + a_2 \delta_s(t) + a_3 \delta_q(t) \tag{4.6}$$

The reason why the old offset $\delta_o(t-1)$ is used is to obtain a smooth transition during transients, i.e. detection of a new offset. Also, using the old value, the output offset will not be as susceptible when the windows detect an offset that deviates from the true offset during shorter periods.

4.3 Parametrisation

In (4.1) to (4.6) the main equations of the algorithm are presented. These equations include tuneable parameters that affect the performance of the algorithm in terms of accuracy and characteristics.

In order to tune the parameters, requirements have been defined and are as follows:

1. The algorithm should be able to determine a new offset in 60 seconds.
2. The algorithm should not oscillate around the current offset.
3. The algorithm has 4 kilobytes memory storage available.

The first requirement is related to the settling time when a new offset occurs (equivalent to a step). The second requirement is necessary for the result to be accurate and stable. The third requirement is a hardware limitation. Although the algorithm in this thesis has been developed in MATLAB no memory or hardware limitations exist. However, as it could be interesting to implement the algorithm on a truck, this memory storage limitation is used as guidance during parametrisation since the algorithm stores samples.

In Table 4.1 the tuneable parameters are listed.

Table 4.1: *Tuneable parameters.*

Parameter	Description
$Resolution$	The bin size of the frequency distribution
a_1	Weighting parameter for previous offset
B_{low}	Lower boundary for changing between offsets
B_{high}	Higher boundary for changing between offsets
w_{slow}	Number of samples in long window
w_{quick}	Number of samples in short window

4.3.1 Resolution

To begin with, the resolution of the bins needs to be determined. The previously mentioned play in the steering system might affect low SWA values, and these low steering input angles will therefore not be manifested on the front wheels.

Also, if the resolution is too high, the SWA samples will be distributed over more bins in the frequency distribution. This results in a MODE that changes more frequent, albeit the changes occur in a small interval.

If instead the resolution is lowered, the SWA samples will be collected in a fewer amount of bins and a more steady MODE can be found. If the resolution is lowered too much the algorithm will experience difficulties in finding smaller offsets in the SAS.

4.3.2 Weighting Parameters

The weighting parameters a_2 and a_3 are automatically determined according to the strategy in (4.4), but a_1 needs to be predefined. The algorithm is designed such that the sum of a_1, a_2 and a_3 equals 1.

A too high value of a_1 results in slow settling time when a new offset has been detected. This is true since the output of the algorithm relies on itself at the previous time stamp and the higher a_1 the more the offset relies on the previous offset. On the other hand, a too low value of a_1 results in a fast settling time which is undesirable as too quick changes might affect other functions that are supposed to receive the SAS offset.

4.3.3 Boundary Parameters

The boundary parameters are needed in order for the algorithm to know if it should weight the quick or slow part. Since the algorithm needs a strategy when weighting the slow or the quick offset, the boundary parameters are used as a lower and an upper threshold.

The boundary parameters are B_{low} and B_{high} with the same unit as the SWA samples. Parameter B_{low} is parametrised to achieve a steady offset output when the absolute difference between the slow and quick offset is low. The idea is that if the quick offset has found a new offset less than B_{low} it will not affect the steering notably and eventually $\delta_q = \delta_s$ as soon as the slow calibration has found the new peak. Since the (simplified) ratio from the steering wheel to the front wheels is 1:20, a boundary value of for example $B_{low} = 1°$ would result in a 1° wrong offset until $\delta_s = \delta_q$. This is equivalent to 0.05° on the front wheels. This tiny offset will most likely not be manifested on the front wheels, due to the play in the steering system. The parameter is also parametrised to manage conditions 3-5 in Section 3.4 by setting it to a value that approximately cover some of the circumstances when these conditions occur.

A proposed value of B_{low} is the resolution in degrees +1°.

The boundary parameter B_{high} is determined as when a notable offset differing from the current is detected. When the absolute difference of the slow and quick offset exceeds this boundary the quick offset is weighted more, enabling a quick calibration. The value is set to a value corresponding to a notable change in the offset.

4.3.4 Window Size Parameters

The size of the windows are determined by the hardware limitations, such as the memory storage available. The harsh memory storage requirement is, as mentioned earlier, only relevant on embedded code but can be used as guidance during development. Every sample within the window size will be stored, in order to calculate the MODE. Increased window size will also have an impact on the run time of the algorithm.

The window size responsible for the quick calibration, w_{quick}, must not be too small as the quick part will oscillate too much, and it will be difficult to find a significant peak. If w_{quick} is too large, the algorithm will not be able to perform a quick calibration. Therefore, balance is needed in order to find a decent parameter value.

If w_{slow} is too small the calibration will not be steady enough, see Figure 4.2 where a test equivalent to Test A in Section 5 is performed. The blue line, (the offset) fails to be steady due to the small window size of w_{slow}. On the other hand, if it is too large the memory requirement will not be fulfilled, the algorithm will be slower in run-time and the idea of that δ_s reaches δ_q will take too long, resulting in a wrong offset as a consequence.

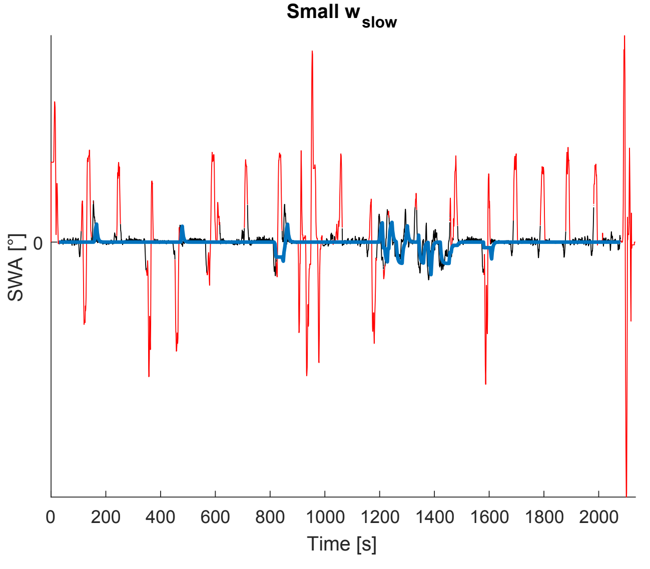

Figure 4.2: *The blue line is the offset determined by the algorithm when a small w_{slow} is used. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.*

A parametrisation has been done where $w_{slow} = w_{quick} * 10$ and w_{slow} holds a maximum of one minute of history when the slow MODE is calculated. The memory limitation is within reach.

5 Results

In this chapter the result of the algorithm is shown in different tests. The grading on the Y-axis has been removed in all plots as it might expose some of the parameter values. The zero degree tick is given and the maximum absolute SWA value is 12°.

5.1 Tests

In Table 5.1 the test cases are shown. Since the data was logged from a calibrated truck the offset in the original data is around zero. Loading the carriage was not possible, hence steps have been introduced in the data corresponding to an offset in the system. A discussion concerning if this is a reasonable method can be found in Section 6.1.2. The steps are aimed to test the different scenarios that may occur and might affect the performance of the algorithm.

Table 5.1: *Different test cases to study the performance of the algorithm.*

Test	**Description**	**Step size**
A	Original data	None
B	Induced step in data	Step $< B_{low}$
C	Induced step in data	$B_{low} \leq$ Step $\leq B_{high}$
D	Induced step in data	Step $> B_{high}$

5.1.1 Test A - Steady State

The first test, Figure 5.1 shows the stationary behaviour of the algorithm. Compared to Figure 4.2, the output of the algorithm is more stable when the parameters are tuned and the offset is detected to be zero almost at all times, which is accurate. However, at $t = 1200$ to $t = 1500$ the offset deviates from zero at three times. As can be seen on the SWA data hardly no segmentation is performed here, and the input on the steering wheel is rather frequent in both directions. This results in an offset that presumably are changing between two peaks as they occur. With larger window size parameters this behaviour can be handled, but with slow offset detection and memory problems as consequences. A lower resolution can also solve the problem, but with reduced accuracy as a consequence. Therefore a better segmentation is needed to remove scenarios when the vehicle is actively turning and the vehicle speed is above the minimum speed.

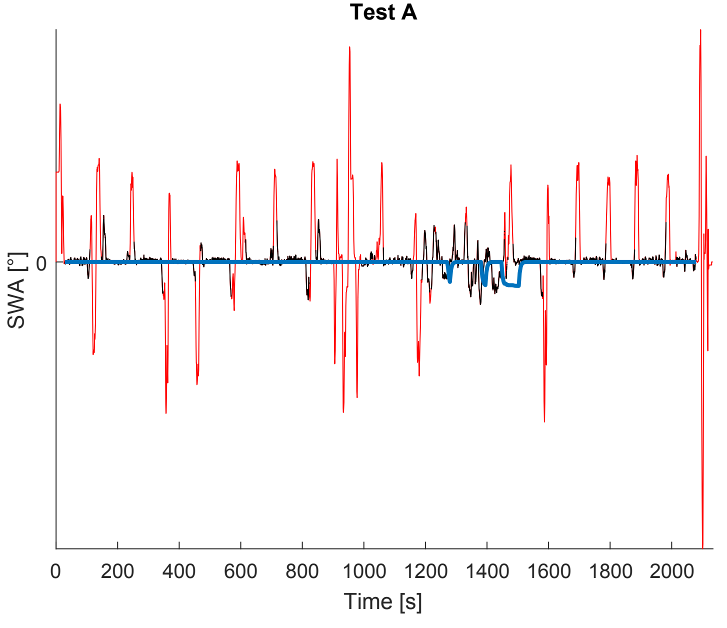

Figure 5.1: Test with original data. The blue line is the offset determined by the algorithm. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.

5.1.2 Test B - Small Step

In this test, the step is below the lower boundary parameter (4.4). Thus, the algorithm will not change its output until the slow part has detected the new offset. This results in a slower calibration time. Since B_{low} is parametrised for very small offset deviations, this longer calibration time is acceptable as the size of an offset smaller than B_{low} is not critical for the functions depending on the correct SWA. It takes just below 60 seconds for the algorithm to calibrate the SAS and deliver an output equal to the step size from when the step has been introduced.

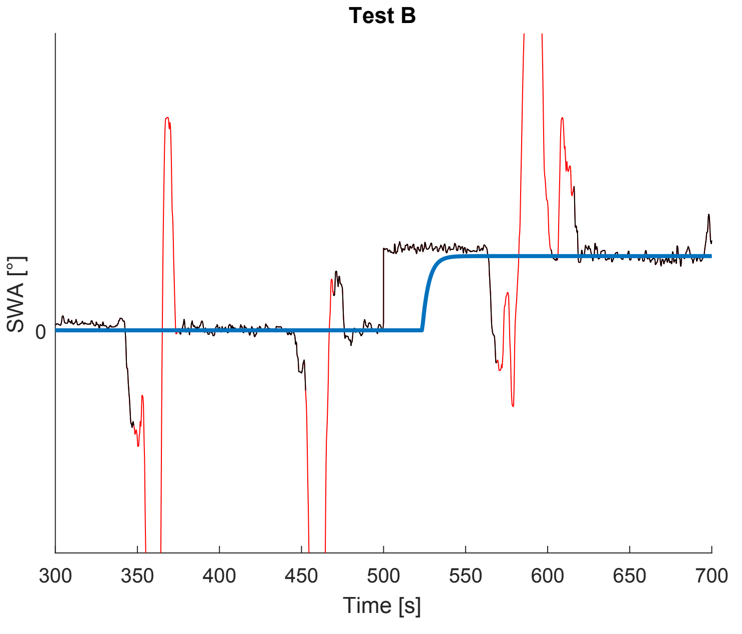

Figure 5.2: *Test with induced offset below B_{low}. The blue line is the offset determined by the algorithm. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.*

5.1.3 Test C - Middle Sized Step

In Test C the step size is above B_{low} and below B_{high} (4.4). When an offset occurs between the boundary parameters it results in this odd shape of the offset. This happens because both δ_s and δ_q are weighted and only δ_q has reached the new offset, and at a certain time this terrace point occurs. As soon as $\delta_s = \delta_q$, the output of the algorithm will begin to reach the accurate offset. The reason why the algorithm weights both contributions is because the quick part has found a new offset that is not too big. If this is only a false alarm, such as a problematic driving condition, the slow part will never reach this level as it is at the true level. Thus, by weighting both contributions, the algorithm hedges towards uncertainties and makes sure that the output error is smaller. If the slow part finds the offset found by the quick part, the scenario in Test C is obtained.

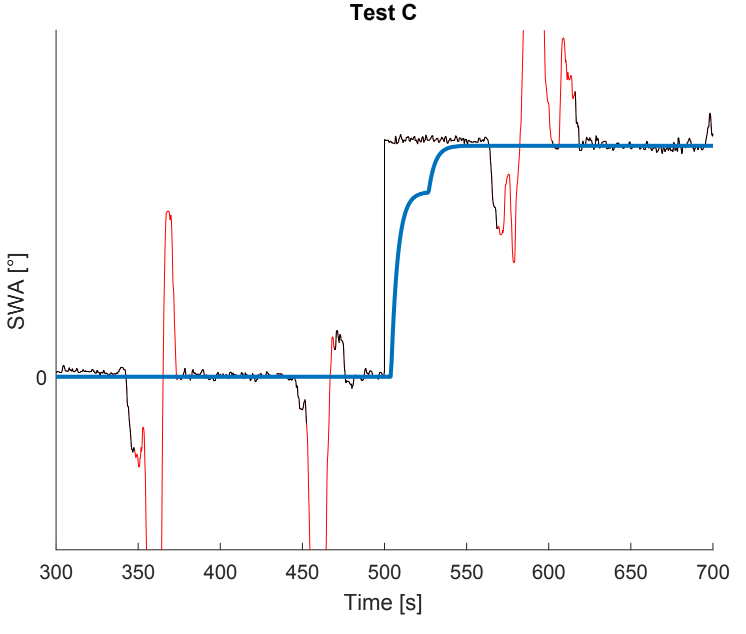

Figure 5.3: *Test with induced offset between B_{low} and B_{high}. The blue line is the offset determined by the algorithm. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.*

5.1.4 Test D - Large Step

The last test includes a step exceeding B_{high} (4.4). In this scenario, the offset found by the quick part is considered to be highly deviating thus the quick part is weighted more to ensure a quick calibration, since the slow part has not detected the new large offset. An offset exceeding B_{high} is of such magnitude that a quick calibration is needed since this large offset might directly affect waiting functions relying on the SAS information. The algorithm succeeds to find the offset in 40 seconds, which is the fastest calibration time of all step cases. The quick calibration time is possible since the algorithm does not need to wait for the slow part, and the quick part's main mission is to find new offsets quickly, which it does.

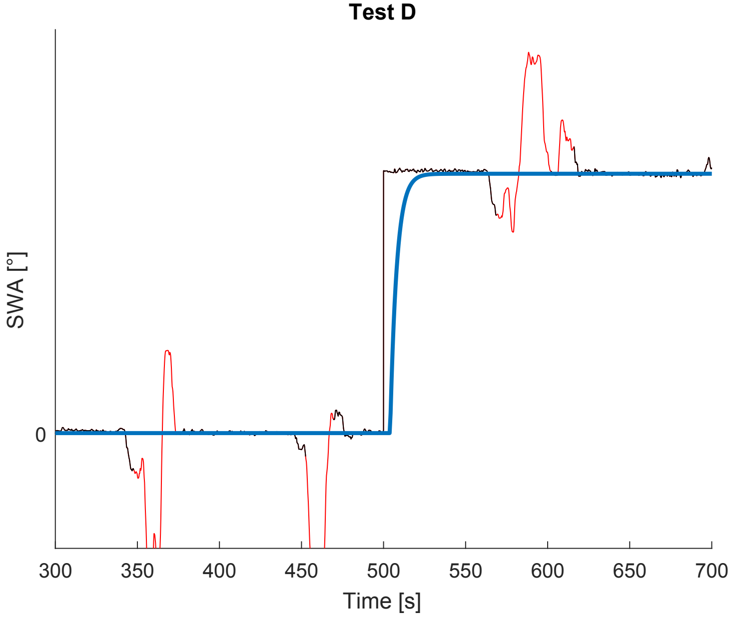

Figure 5.4: *Test with induced offset exceeding B_{high}. The blue line is the offset determined by the algorithm. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.*

5.2 Detailed Results

In Figure 5.5 the accuracy of the algorithm can be studied when the vehicle is driving nearly straight ahead at four periods. The spikes in the figure correspond to when the vehicle is turning and most parts of the turns with high SWA values are successfully segmented. The SAS signal is never constant but oscillates mostly around zero. Since the resolution of the algorithm is the size of integers, all SWA samples in the interval of [X-0.5, X+0.5] are lumped into the offset angle X. The reason why this resolution is used is discussed in Section 4.3.1. Studying the figure, a certain overweight for positive SWA samples can be seen. However, should the offset be 0.1° it would never be manifested due to the nonlinearites in the steering system.

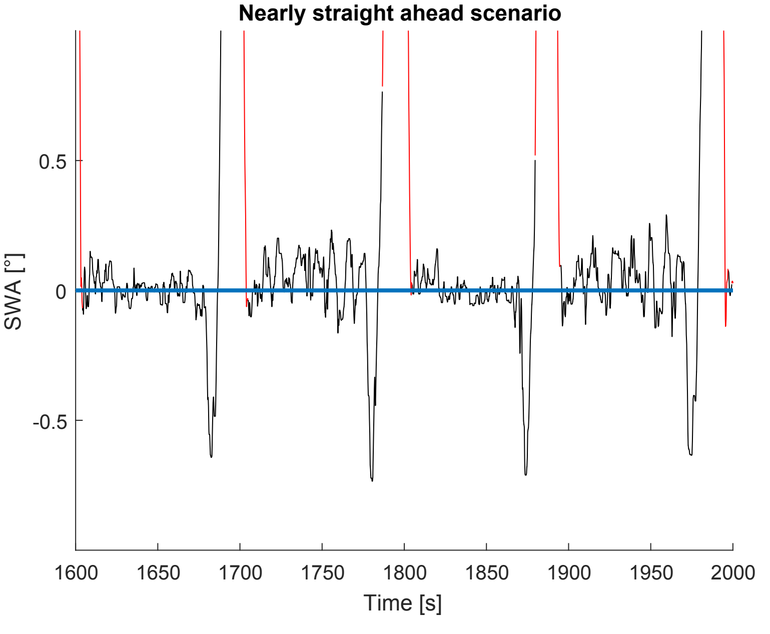

Figure 5.5: *The blue line is the offset determined by the algorithm. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.*

Studying Figure 5.6 the problematic behaviour in Test A is in focus. In the top figure the parameters have the same values as in Test A-Test D. As can be seen, the segmentation is not successful, as the offset (blue line) is shifting between two values. In the bottom figure, the minimum speed has been increased to 40 km/h, which was the highest appropriate minimum speed to use for the segmentation, as mentioned in Section 3.5. This results in an algorithm that handles the problematic behaviour by supplying the correct output at all times, with less activity as a consequence.

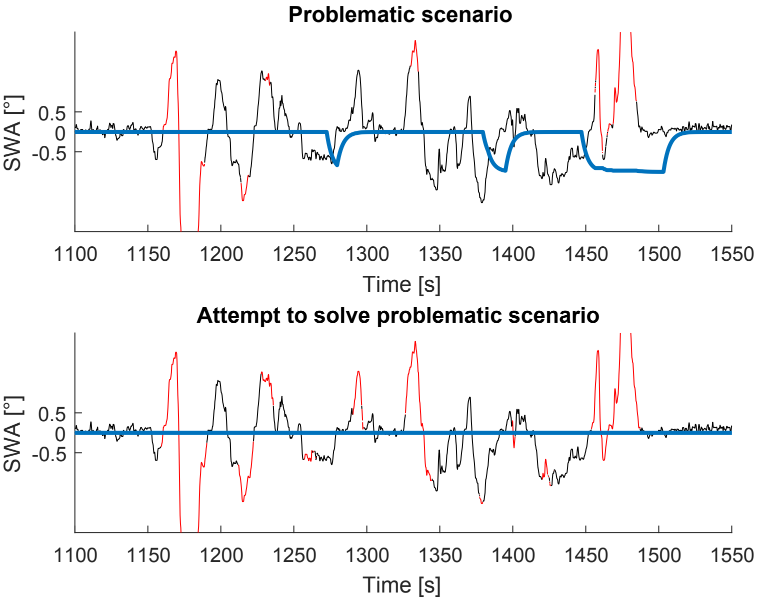

Figure 5.6: *The blue line is the offset determined by the algorithm. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.*

5.3 Calibration Time

From the tests above the calibration time varies depending on how big the offset is compared to B_{low} and B_{high}. Other factors that affect the calibration time are, for example, loading and unloading. If the carriage of the truck is loaded, with an offset introduced as consequence, the algorithm will be active only when the speed is above the minimum speed. This will then affect the calibration time. However, for low speeds, offsets are not as critical since ranging and cornering occur more frequent and the fact that the driver is countersteering will not be as notable.

Another factor that affects the calibration time is when the offset occurs. Since the moving windows might hold only straight ahead samples and the offset is within the boundary parameters (4.4), the offset will be detected only when both the quick and slow part has found the new peaks which depends on the window sizes.

In Table 5.2 it can be seen that from the results above, the calibration is indeed fastest for the quick calibration and slowest for the slow calibration which is intended. Test A is not interesting concerning calibration time.

Table 5.2: Calibration time results.

Test	Calibration	Characteristics
D	Quick Calibration	Fastest
C	Combination	Second
B	Slow Calibration	Slowest

It could be interesting to study the performance and calibration time of the algorithm if the output is not filtered, allowing for instantaneous changes when an offset occur. In Figure 5.7 such configuration can be seen, with a test equivalent to Test D and same parameters. When the quick part has detected an offset the output of the algorithm changes instantaneously to that value which could be troublesome to handle by functions that need the offset information. Therefore, filtering the output signal is recommended.

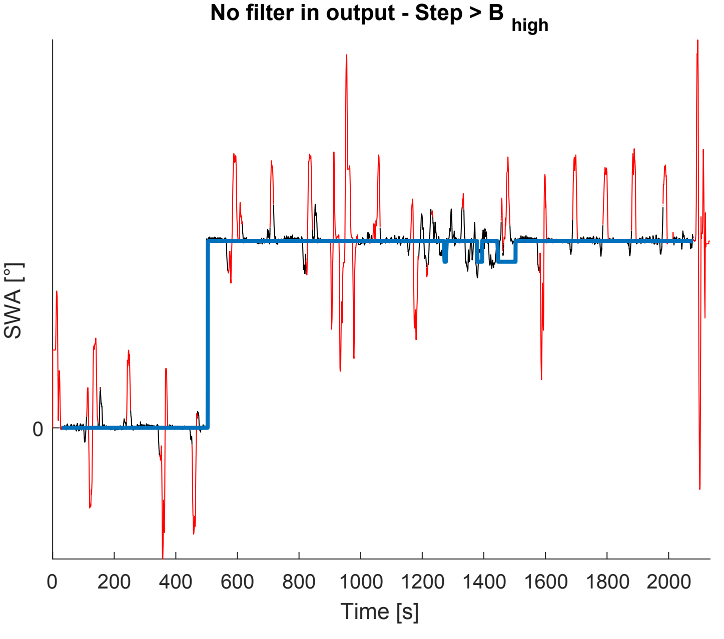

Figure 5.7: *Test with induced offset exceeding B_{high}. The blue line is the offset determined by the algorithm when the output is not filtered. The black and red lines are the SWA samples where the black line corresponds to the algorithm being active and red when the algorithm is not active.*

6 Summary

The developed algorithm is combining the SWA MODE of two windows with different length and is only active when the longitudinal vehicle speed exceeds the minimum speed. The combination part makes use of a simple filter and a function to choose which MODE that should be weighted. Furthermore, the function only uses the SAS sensor and the longitudinal vehicle speed information thus without these information sources it can not function properly. As of now the resolution of the determined offset is the size of integers in degree since it was the requirement.

6.1 Discussion

In this section the performance of the algorithm is discussed based on the results in Chapter 5. The data used for the algorithm and the data manipulation for the tests are discussed and a brief discussion and analysis of the memory requirement is presented.

6.1.1 Performance

Regarding the accuracy of the algorithm the exact offset with current parametrisation is almost impossible to achieve and the output could always differ $\pm 0.5^{\circ}$ from the true SAS offset, see Figure 5.5. This is, as previously mentioned, an accuracy good enough due to the nonlinearites in the steering system. It is possible to achieve a finer resolution by a different parametrisation.

Studying the results in Chapter 5, Test A, see Figure 5.1, shows a flaw in the algorithm as it has problem when the vehicle is actively not driving straight ahead at $t \approx 1200$ to $t \approx 1500$. This partly depends on the resolution and window sizes.

With an increased resolution or increased window size this behaviour could be handled. It also depends on that the slow part does not need a significant peak to detect an offset. Should the slow part be configured similar to the quick part this unwanted effect could possibly be handled. However, this would have negative impact on the computational time. It might also be that the segmentation part needs further development to cover such scenarios where the simplest solution is to increase the minimum vehicle speed, see Figure 5.6. The disadvantage is that the algorithm will be less active, hence the algorithm would be useless in low speed applications.

In Test B-Test D an inaccurate offset output is delivered during the calibration time, although decreasing as the output reaches the true offset. Even if this characteristic is desired, since a binary switching is not wanted as in Figure 5.7, it can still be problematic as functions depending on an accurate offset may rely on this inaccurate offset during the calibration time. A plausible solution could be to add information how wrong the offset is at each time stamp, since the targeted offset value is known by the algorithm. Besides the wrong offset during the calibration time, the algorithm manages to detect all kinds of offsets with a desirable calibration time.

6.1.2 Data

The algorithm has only been tested on one set of data with an offset around zero. Instead of introducing steps to imitate offsets a more extensive data acquisition could have been pursued. Data logs including loading and unloading, and the problematic driving conditions would have been interesting to study, in order to cover the different scenarios naturally to test the performance of the algorithm more extensively.

The manipulation of the data in Test B-Test D has been done to mimic an offset in the SAS. Since the real offset is known to be 0°, creating a step in the data is a simple yet an efficient way to test the algorithm for different offset sizes. As long as the offset appears similarly in real life, which they should do, the steps in Test B-Test D are sufficiently good to validate the algorithm. The certain moment in time when a step should occur to mimic an offset is unknown but at the same time irrelevant since the aim of the tests is to find the offset, which the algorithm successfully does.

6.1.3 Memory Allocation

By creating corresponding frequency distributions for the slow and quick windows it is possible to save values in the windows corresponding to the index in the frequency distribution instead of the raw SWA samples. When a new value is added to a window the corresponding index in the frequency distribution is incremented. When a value is removed from a window the corresponding index in the frequency distribution is decremented. This method results in four arrays that will allocate memory: two moving windows and two frequency distributions. The advantage is that only unsigned integers of 8 or 16 bits need to be stored instead of the raw SWA values which are of double type, i.e. 64 bits. Also, the MODE operation can efficiently be done in the frequency distributions by finding the index with the maximum value and then cast the index value to a signed integer to obtain the corresponding offset in each window.

To further explain the strategy, Table 6.1 is used. The data in the table is mocked data. The frequency distributions H_{slow} and H_{quick} are of length 201, and can therefore hold SWA samples in the range $[-100°, 100°]$. Thus only unsigned 8 bits integers need to be stored in the windows w_{slow} and w_{quick} since the windows only hold the corresponding element indexes in the frequency distributions. The length of $w_{slow} > 256$, thus H_{slow} needs to hold unsigned 16 bits integers, since the max value an element in H_{slow} could obtain is the length of w_{slow}. The same principle goes for the quick part, and since the length of $w_{quick} < 256$ the corresponding frequency distribution only needs to hold unsigned 8 bits integers.

Table 6.1: *Memory allocation strategy (with mocked data). The byte calculations are approximate.*

Parameter	Length	Type	Bytes
w_{slow}	10000	uint08	10000
w_{quick}	100	uint08	100
H_{slow}	201	uint16	402
H_{quick}	201	uint08	201
Sum			10703

The current parametrisation of the algorithm's heavier parameters allocates 3.7 kilobytes. There are still space for the constant parameters to achieve the memory requirement.

6.2 Conclusion

The developed algorithm performs well on the tested data. This thesis shows that it is possible to use a statistical approach when determining the SAS offset. The algorithm succeeds in determining the offset regardless of being stationary or temporary. The algorithm itself is rather simple and has no vehicle specific parameters, thus it works in almost every vehicle as it only uses two common sensor input signals. The algorithm requires storage of samples and the memory requirement is fulfilled. However, the algorithm may not function properly in all applications, such as mining vehicles, where speeds usually are low, ranging occurs more frequently and the road curvature is changing.

Also, the algorithm will presumably perform different for different drivers. A driver that steers the vehicle steady will most likely result in a more reliable and steady offset, compared to a driver that frequently turns the steering wheel.

It cannot be concluded how much banking and crosswinds affect the algorithm as the data was insufficient in covering all problematic driving conditions.

6.3 Future Work

Implementing the algorithm on a truck is highly interesting to study the performance. The development environment can then be used as a benchmark to further facilitate the hardware implementation. The written code needs to be translated to C-code and some of the built-in MATLAB functions need to be rewritten. The storage requirement would be critical, thus the memory allocation needs consideration. To reduce bus load the algorithm can be configured to still receive SWA samples for each sample time corresponding to the raw SAS signal, but perform calculations for an arbitrary larger sample size.

With an implementation on a truck, questions regarding if the algorithm works as expected, how it works delivering the offset to other functions and how it performs in different applications could be answered. These answers could then be used to further improve the performance of the algorithm. It will also be possible to validate that the tests performed in this thesis are realistic. If they are, the future development of the algorithm may be time saving and cost efficient as testing can be performed in the development environment.

Secondly, parameter tuning can be done differently to acquire desired characteristics aimed for a certain function. If for example a higher resolution is needed the window sizes would presumably need to be changed as there would be more bins in which the SWA samples are stored resulting in more than one peak in the frequency distributions. Depending on when a slow or quick calibration is needed, the boundary parameters can be tuned to achieve the desired behaviour.

Then, an interesting approach would be to use machine learning to be able to detect driving scenarios and to eliminate the vehicle speed sensor dependency. Much more data would be needed during development, covering all problematic driving conditions as well as different kinds of applications. This could possibly solve the problem of supplying a steady output when the vehicle is not driving straight ahead, as mentioned earlier.

Bibliography

[1] Jörg Grotendorst Andreas Zell. Method and device for determining a steering angle offset, 2013. Cited on page 2.

[2] Sean Bennett. *Heavy Duty Truck Systems*. Cengage Learning, Boston, MA, 6 edition, 2015. ISBN 9781305073623. Cited on page 9.

[3] Peter Fejes. Estimation of steering wheel angle in heavy-duty trucks. Master's thesis, Uppsala University, 2016. Cited on pages 3 and 5.

[4] David Freedman. *Statistics*. W.W. Norton, New York, 1998. ISBN 0-393-97083-3. Cited on page 10.

[5] Manfred Harrer. *Steering handbook*. Springer, Cham, 2016. ISBN 978-3-319-05448-3. Cited on pages 5, 9, and 16.

[6] Xiaoyu Huang and Junmin Wang. Identification of ground vehicle steering system backlash. *Journal of Dynamic Systems, Measurement, and Control*, 135(1), 2012. doi: 10.1115/1.4007558. Cited on page 6.

[7] Haik Kalantarian, Costas Sideris, Tuan Le, Christing King, and Majid Sarrafzadeh. A framework for probabilistic segmentation of continuous sensor signals. In *Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, 2016. Cited on page 10.

[8] Willy Killer and David VanderLugt. Continuous correction for steering wheel angle offset, 2013. Cited on page 2.

[9] D.S. Turner W. Brilon L.E. Hall, R.D. Powers and J.W. HaLl. Overview of cross section design elements. In *International Symposium on Highway Geometric Design Practices*, January 1998. Cited on page 9.

[10] Martin Meywerk. *Vehicle Dynamics*. Wiley, Chichester, West Sussex, 2015. ISBN 9781118971376. Cited on page 9.

[11] Richard Sjölund and Nicklas Vedin. Steering system modelling for heavy duty vehicles. Master's thesis, Linköping University, 2015. Cited on page 5.

[12] J. Y. Wong. *Theory of ground vehicles*. John Wiley, New York, 2001. ISBN 0-471-35461-9. Cited on page 8.