



University of HUDDERSFIELD

University of Huddersfield Repository

Berkenbusch, Jan

Development of a control system for automated cab roof deflectors for articulated trucks

Original Citation

Berkenbusch, Jan (2015) Development of a control system for automated cab roof deflectors for articulated trucks. Masters thesis, University of Huddersfield.

This version is available at <http://eprints.hud.ac.uk/id/eprint/29076/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

Development of a control system for automated cab roof deflectors for articulated trucks

Jan Berkenbusch

Director of Studies: Prof. Rakesh Mishra

A thesis submitted to the University of Huddersfield in fulfilment of the requirements for the degree of Master of Science by research

November 2015

University of Huddersfield

School of computing and engineering

Abstract

The adverse effects of emissions from the global transport industry on the environment have become increasingly apparent, with significant emphasis placed on their reduction through various initiatives. In the UK, Heavy Goods Vehicles (HGV's) are used to transport 68 % of all goods. The majority of trucks are articulated (Truck and Trailer) – rather than rigid vehicles. Surveys show that within the UK in 2014, HGV's travelled approximately 18,769 million kilometres ⁽¹⁾. The CO₂ emissions from transport industry are estimated to account for 25% of entire emissions, with HGV's responsible for 23 % of transport emissions ^(2, 3). Any improvement in fuel economy within transport industry will thus have direct bearing on management of emissions.

Outer dimensions of HGV's are regulated by the local and European laws, with the design of these vehicles optimized to maximise cargo volume rather than aerodynamic efficiency. The goal of this research was to reduce fuel consumption of articulated HGV's by reducing the aerodynamic drag by modifying flow field around the vehicles ^(4, 5). A number of post market add-on devices for commercial vehicles have been adapted as standard equipment. Whilst some of these devices like the cab roof fairing have become popular, others like the base bleeding devices have been largely discarded by the industry due to practical limitations like obstruction caused in normal operations e.g. loading.

Studies have shown that streamlined matching of trailer shape and cab roof deflector for smooth airflow can reduce fuel consumption of articulated trucks by approximately 7.6% ⁽⁶⁾. Researchers found out that just the positioning of a cab roof deflector (CRD) can influence the fuel consumption of an articulated truck by about 4 % ⁽⁷⁾. Existing cab roof deflectors are optimised for a single trailer height. However in practice HGV's will be required to operate with a range of trailer heights. To maintain efficiency it would require a change in the deflector position each time the trailer size is altered. Furthermore it is required to respond to changing crosswind conditions with changing CRD positioning for maximum efficiency

Through this research project a control system for automated dynamic CRD adjustment has been developed. Its concept can be employed for any combination of truck and trailer. The control system has been developed to always adjust the CRD to the optimum position for reducing aerodynamic drag. The reduction in aerodynamic drag is directly related to the reduction of fuel consumption.

Declaration

- i. The Author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the “copyright”) and he has given it The University of Huddersfield the right to use such Copyright for any administrative, promotional and/or teaching purposes.

- ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the University Library. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

- iii. The Ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Contents

Abstract.....	2
Declaration.....	3
Contents	4
List of Figures.....	6
List of Tables.....	9
Nomenclature.....	10
1. Introduction	12
1.1 Motivation.....	19
1.2 Research Aim.....	20
2. Literature review.....	21
2.1 Aerodynamics of articulated trucks	22
2.2 Specific research objectives.....	34
3. Development of a multihole probe system for quantification of wind angle in the vicinity of a vehicle.....	36
3.1 Pitot Tube.....	37
3.2 Three-hole probe.....	38
3.3 Five-hole probes	39
3.4 Selection of pressure sensors	42
3.5 Five hole probe calibration	45
3.5.1 Step wise description for the calibration of a five hole probe.....	51
3.6 Calculation of wind information from a single set of five hole probe pressure readings	53
3.6.1 Step wise description for the calculation of wind information from a set of five hole probe pressure readings	54
3.7 Validation of the calibration results	55
3.8 Conclusion	58

4. Development of a control circuit and a sensor array for the automation of an automated cab roof deflector	60
4.1 Selection of distance sensors	63
4.2 Control system setup.....	65
4.3 Conclusion	67
5. Software Development	69
5.1 Software architecture	71
5.1.1 Microcontroller configuration	71
5.1.2 System initialization.....	74
5.1.3 Main programme	75
5.2 Conclusion	91
6. Evaluation of the developed system	93
7. Conclusion.....	100
7.1 Review and attainment of objectives	101
7.2 Original contribution of the work	102
7.3 Major conclusions of the work	103
7.4 Recommendation for future work.....	103
8. References	107
Appendices.....	112
A1 Five hole probe calibration parameters.....	112
A2 Control system main circuit board.....	117
A3 Table with compiled CFD Results	119
A4 Source Code.....	122
A5 DVD with CFD results files	160

List of Figures

Figure 1-1: Dimensions of an articulated truck (schematic, not scaled).....	14
Figure 1-2: Power required to overcome the aerodynamic drag and rolling friction as a function of travel speed for a typical class 8 truck trailer [5]	15
Figure 1-3: The fuel saving percentage vs. drag reduction percentage for different scenarios [4].....	16
Figure 1-4: Flow separation on an articulated truck (schematic) [6].....	18
Figure 2-1: Matching the cab and container flow field for optimum aerodynamic interference [8].....	24
Figure 2-2: Example curves of normalized drag coefficient as a function of tractor/trailer gap for several vehicle configurations [7]	26
Figure 2-3: Sketch of the cross flow vortex trap gap treatment device installed on the trailer front face. [13]	29
Figure 2-4: Different combinations of fairing on the baseline semi-trailer truck model [16] .	31
Figure 2-5: Drag coefficient as a function of speed for different test configurations at yaw angle $\beta = 0^\circ$ [16].....	32
Figure 3-1: Principle of a pitot tube probe [21]	37
Figure 3-2: Schematic of a cylindrical two dimensional probe tip [22]	38
Figure 3-3: Wind vectors on a truck in crosswind condition [7]	40
Figure 3-4: Labelled picture of the used five hole probe.....	41
Figure 3-5: Nomenclature of a five hole probe and setup of a five hole probe sensor system	42
Figure 3-6: Beaufort scale for wind velocities.....	43
Figure 3-7: Pressure sensor board with five mounted sensors type MPXV7002	44
Figure 3-8: Five hole probe mounted on the traverse in the wind tunnel	44
Figure 3-9: Wind tunnel setup for five hole probe calibration.....	45
Figure 3-10: Picture of the wind tunnel centre flow domain element (not scaled).....	46
Figure 3-11: Schematic of the wind tunnel centre flow domain element (not scaled)	47
Figure 3-12: Sector Map created from the calibration data	49

Figure 3-13: Section of calibration raw data.....	55
Figure 3-14: Accuracy for five hole probe calibration in pitch direction	56
Figure 3-15: Accuracy for five hole probe calibration in yaw direction.....	57
Figure 3-16: Accuracy for five hole probe calibration in wind magnitude.....	58
Figure 4-1: 3D model of the used deflector	61
Figure 4-2: Schematic of the setup for the automated cab roof deflector system.....	62
Figure 4-3: Picture of the setup for the automated cab roof deflector system.....	62
Figure 4-4: Deflector height per <i>vout</i> of the used Sensor.....	65
Figure 4-5: PCB with H-Bridge motor driver VNH3SP30-E.....	66
Figure 4-6: Picture of the developed main controller circuit board.....	68
Figure 5-1: Screenshot of AVR Studio development platform	70
Figure 5-2: USB to ISP interface mySmartAVR lite	71
Figure 5-3: circuit schematic of the controller board.....	72
Figure 5-4: Controller software main routine flowchart.....	76
Figure 5-5: Controller software for finding the sector of the sensors dataset flowchart	79
Figure 5-6: Controller software wind information calculation flowchart.....	80
Figure 5-7: Mesh independence simulation results	81
Figure 5-8: Surface representing optimum deflector positioning for 4.2 m reference height .	84
Figure 5-9: Surface representing optimum deflector positioning for 4.6 m reference height .	84
Figure 5-10: Surface representing optimum deflector positioning for 4.8 m reference height	85
Figure 5-11: Spots of positioning surface for 4.8 m sector 1	86
Figure 5-12: Equation system for deflector optimum height 4.8 m section	86
Figure 5-13: Nomenclature of sectors in the positioning table for 4.8 m trailer height	86
Figure 5-14: Controller software optimum deflector position calculation flowchart	89
Figure 5-15: Controller software deflector position adjustment flowchart	91
Figure 6-1: Schematic of the cab roof deflector setup.....	94
Figure 6-2: Test setup of the automated cab roof deflector system	95

Figure A 1: Five hole probe calibration parameters for Sector 1 112

Figure A 2: Five hole probe calibration parameters for Sector 2 113

Figure A 3: Five hole probe calibration parameters for Sector 3 114

Figure A 4: Five hole probe calibration parameters for Sector 4 115

Figure A 5: Five hole probe calibration parameters for Sector 5 116

Figure A 6: Control system main circuit board schematic 117

Figure A 7: Control system main circuit board layout 118

Figure A 8: CFD results for a trailer reference height of 4.2 m 119

Figure A 9: CFD results for a trailer reference height of 4.6 m 120

Figure A 10: CFD results for a trailer reference height of 4.8 m 121

List of Tables

Table 3-1: Overview goodness of fit for calibration parameters	52
Table 4-1: H-Bridge modes of operation	66
Table 5-1: Deflector heights from CFD results for aerodynamically optimum positioning in m	83
Table 5-2: Equations for optimum deflector height for 4.8 m trailer height.....	87
Table 5-3: Equations for optimum deflector height for 4.6 m trailer height.....	87
Table 5-4: Equations for optimum deflector height for 4.2 m trailer height.....	87
Table 6-1: Test results for 4.8 m for changing wind angles (in m)	97
Table 6-2: Test results for 4.6 m for changing wind angles (in m)	97
Table 6-3: Test results for 4.2 m for changing wind angles (in m)	98

Nomenclature

A	=	projected frontal area of a truck in aerodynamic models (m ²)
ADC	=	analogue to digital converter
ADC_{PS_n}	=	output voltage of pressure sensor P_n converted into an ADC value
a_{0-14}	=	calibration parameter for five hole probe calibration
CFD	=	computational fluid dynamics
CRD	=	cab roof deflector
C_D	=	shape based drag coefficient
C_{P_α}	=	pressure coefficient with respect to vertical axis
C_{P_β}	=	pressure coefficient with respect to horizontal axis
$C_{P_{static}}$	=	pressure coefficient with respect to static pressure
$C_{P_{total}}$	=	pressure coefficient with respect to total pressure
D	=	denominator
F_D	=	drag force (N)
f	=	function (generic)
HCV	=	heavy commercial vehicle
h	=	height of the top edge of the cab roof deflector (m)
IR	=	infrared
MCU	=	microcontroller unit
$Offset_{PS_n}$	=	ADC offset value for pressure sensor P_n
\bar{P}	=	averaged pressure (Pa)
P_{1-5}	=	pressures on sensing holes of a five hole probe (Pa)
P_{center}	=	pressure on the centre hole of a three hole probe (Pa)

$P_{dynamic}$	=	dynamic pressure (Pa)
P_{left}	=	pressure on the left hole of a three hole probe (Pa)
P_{right}	=	pressure on the right hole of a three hole probe (Pa)
P_{static}	=	static pressure (Pa)
P_{total}	=	total pressure (Pa)
PCB	=	printed circuit board
PWM	=	Pulse width modulation
Q	=	pressure normalization factor
s	=	scalar factor in vector algebra
t	=	scalar factor in vector algebra
v	=	wind velocity (m/s)
$V_{ref, max}$	=	maximum ADC reference voltage
V	=	resulting wind vector in crosswind situations
V_w	=	wind velocity in crosswind situations
V_{τ}	=	truck velocity in crosswind situations
α	=	pitch angle relative to probe tip centre axis ($^{\circ}$)
β	=	yaw angle relative to probe tip centre axis ($^{\circ}$)
ρ	=	density of air, 1.225 kg/m ³
Δh	=	CRD position difference from actual position to optimum position (cm)

Chapter 1

Introduction

This chapter introduces the Heavy Commercial Vehicles (HVC) and add-on devices which have been used to streamline the airflow around a HCV. The airflow is streamlined to minimize the drag force and consequently reduce fuel consumption of HVCs. This chapter also highlights the basic concept of HCV aerodynamic optimization and defines the research aim for this project.

1. Introduction

Over the last few decades fuel consumption has become a major concern in the automotive industry, especially in high-mileage sectors such as freight and commercial transport companies. Fuel consumption is a major economical factor to the transport companies and also a major environmental factor. Hence the automotive industry needs to improve their energy efficiency on Heavy commercial vehicles (HCV).

HCV were seen as storage on wheels for a long time. This led to designs which are dominated by the idea of maximizing the amount of transported goods per journey. Less emphasis was given towards designing a fuel efficient HCV.

The road transport industry in UK consists of both single bodied small delivery vehicles and articulated trucks consisting of multiple bodies. They are further classified in two categories i.e. articulated trucks and rigid trucks.

Within HCV, the term articulated truck describes a combination of a tractor vehicle with an attached trailer. These vehicles are commonly used for transportation of goods across the country/s. In 2013, 68% of all goods were transported by roads in UK [1] with a significant part being serviced by articulated trucks. Short and local transportation of goods are mostly handled by smaller rigid trucks with fixed loading space and capacity of up to 32 tons maximum gross weight. Articulated trucks are mostly used for long distance transportation of goods with the loading capacity of up to 45 tons.

The dimensions for these HVC are regulated by European and local laws to make the semi-trailers interchangeable with any tractor vehicle. The permissible maximum dimension of trucks in the United Kingdom are of 2.55 m width for any kind of vehicle. Vehicle length is limited to 12 m for rigid trucks and 16.5 m maximum length for articulated trucks [2]. While the geometry of rigid trucks is simple with only the length of the driver cab and the length of the cargo area being a factor, the geometry of articulated trailers is more complicated. An articulated truck is merged by two separate units of truck and trailer. To create interchangeability of trucks and trailers common design guidelines have been developed. Trailer swing radius, distance of the kingpin and the cabs rear wall as well as safety clearance between the swinging trailer and the cab have been standardized. The combination of these factors have led to a general agreement on the overall design of articulated trucks, which is

shown in Figure 1-1. The length of the trailer is 13.6 m, the cab length about 2.0 m. This leaves a gap of 0.9 m between cab and trailer for each standardized semi-trailer vehicle [3].

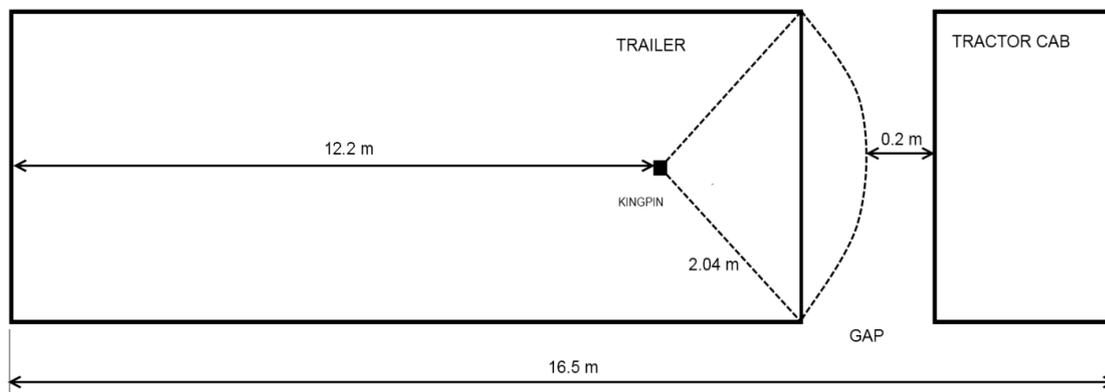


Figure 1-1: Dimensions of an articulated truck (schematic, not scaled)

In any vehicle fuel consumption can be defined as the conversion of stored chemical energy in form of fuel into mechanical energy by a combustion engine. The mechanical energy generated by the combustion engine is applied to the wheels of the vehicle to generate a tractive force. The tractive force must be higher or equal to the resistive forces in order to accelerate the vehicle or sustain in steady motion.

Fuel consumption requirements of trucks on a trip can be split into four major factors: acceleration, drive train losses, rolling resistance and aerodynamic drag.

To accelerate an object the inertia has to be overcome by the applied force. This also applies for the acceleration of vehicles such as HCV. The force needed to accelerate a HCV at low vehicle velocity is mainly dependent on the mass of the HCV and not dependent on the aerodynamics of the vehicle. Thus the fuel consumption for accelerating a HCV at low vehicle velocity is mainly related to the mass of the HCV and not dependent on the aerodynamics of the vehicle.

Drive train losses incorporate heat production in the engine and friction losses which are inherent with the technology of combustion engine vehicles. This also includes power consumption of secondary electrical devices that are powered via the lighting dynamo. Secondary electrical components are, for example, air conditioning, navigation, radio and advanced drivers assistance systems.

In constant speed condition, such as driving on a motorway, acceleration is not taking place. Rolling resistance is a force related to the friction between the trucks tyres and the road. This factor is about constant with slight fluctuation due to different tyre pressures, tyre rubber compound and tyre wear level as well as road surface properties. The rolling resistance factor is usually specified by the tyre manufacturer.

The fuel consumption for overcoming aerodynamic drag is about 38.2 % if the trucks journey mainly consists of steady speed sections with low height difference and low acceleration periods. In that condition the further fuel consumption is distributed as follows: 35 % to overcome rolling resistance, 18.5 % on transmission and electrical accessories and 8.3 % on acceleration [4]. The longer steady speed sectors on a journey are, the more fuel consumption can be related to aerodynamic drag.

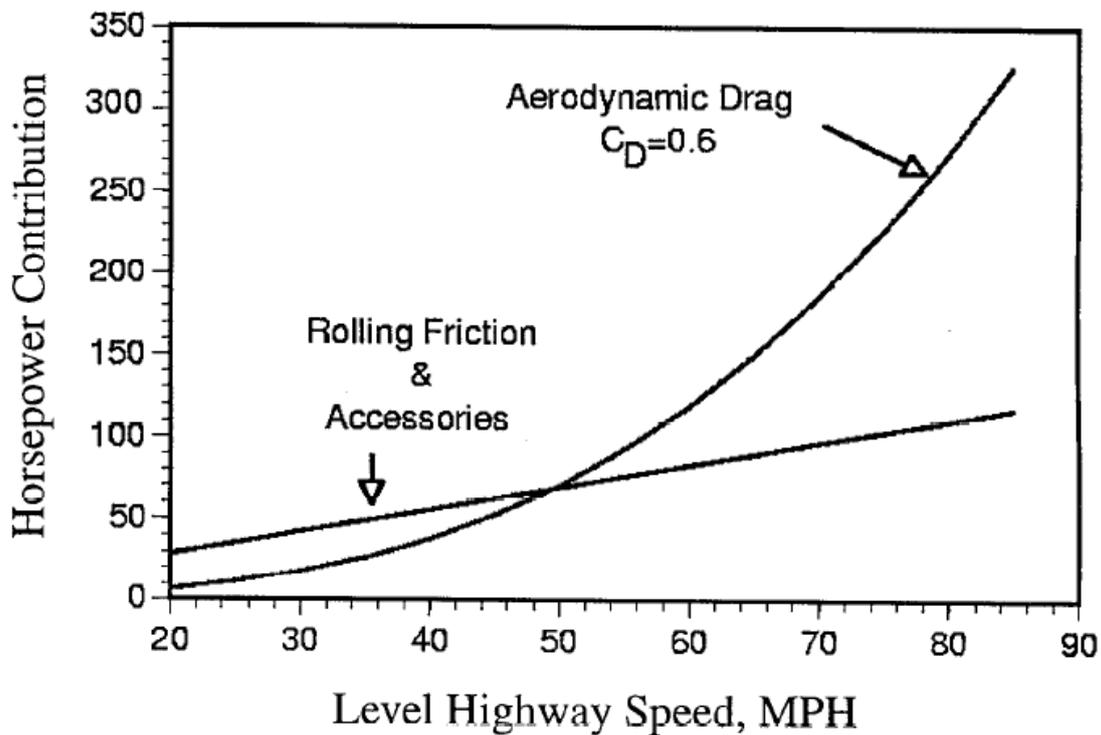


Figure 1-2: Power required to overcome the aerodynamic drag and rolling friction as a function of travel speed for a typical class 8 truck trailer [5]

Figure 1-2 shows the required amount of power to overcome aerodynamic drag and rolling resistance as a function of speed for a class 8 truck-trailer unit with an associated drag coefficient of 0.6. The data shown by Figure 1-2 was obtained by McCallen et al [5] in 1999

and it is claimed at a speed of 70 miles per hour. In that condition around 65 % of the total energy produced by the engine is used to overcome the aerodynamic drag force.

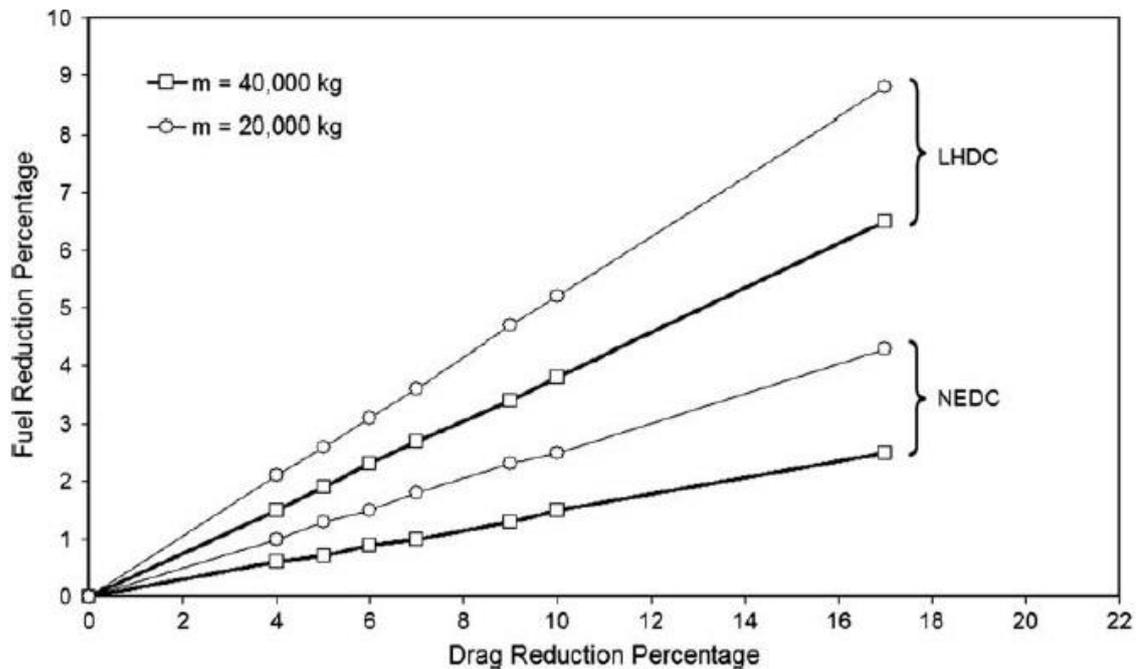


Figure 1-3: The fuel saving percentage vs. drag reduction percentage for different scenarios [4]

With aerodynamic drag being the major part of fuel consumption on long journeys effort in reduction of aerodynamic drag offers rewards in reduction of fuel consumption. Figure 1-3 shows the reduction in fuel consumption that can be achieved by reducing aerodynamic drag as stated by Mohamed- Kassim et al.[4]. The figure shows the relation of aerodynamic drag reduction and fuel consumption reduction for two predefined driving cycles. The LHDC (Long Haul Driving Cycle) which consists of an acceleration sequence of 60 mph in the beginning and a breaking sequence in the end of a 20 minute journey. NEDC (New modified European Driving Cycle) that contains four sequences of stop and go representing urban driving and one 60 mph sequence of 6 minutes. It shows that reduction of aerodynamic drag on HCV has higher effect on fuel saving with extending steady speed motorway sequences on a HCV journey. A comparing of the LHDC and NEDC fuel consumption reduction results shows that the same level of drag reduction results in twice as much fuel reduction for the motorway journey compared to the urban driving cycle. For a truck with a mass of 40 tons and a reduction on aerodynamic drag of 10% a resulting fuel saving of about 5% can be expected.

In general, the shape of a vehicle influences the aerodynamic force which is created by the flow field on and around the vehicle. The resulting aerodynamic pressures and velocities in the flow field influence the magnitudes and directions of forces and moments acting on the vehicle. These forces and moments can be represented in form of drag force, lift force, side force, rolling moment, pitching moment and yawing moment. The drag force created by the flow field is directed opposite to the direction of travel of the vehicle and directly influences the fuel consumption. The other components of the aerodynamic force system mainly influence the stability of the vehicle [6]. The drag force is defined as:

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

ρ represents the density of the fluid. v represents the fluids velocity relative to the object velocity. C_D represents the drag coefficient based on the objects shape that is in the stream. A represents the projected area facing the stream. Equation 1 shows that the drag force increases linearly as the projected frontal surface increases. The drag force also increases linearly as the density of the fluid or the shape based drag coefficient C_D increases. Increasing the velocity of the vehicle increases the drag force by the square of the velocity. Instead of aerodynamically optimizing their vehicles the transport companies focussed on moving as many goods as possible to reduce the cost per transported good.

Over the years the requirement of interchangeability of trucks, trailers, train cargo containers and shipping containers resulted into standardized designs rules for these components. These design rules have to be respected for any aerodynamic improvement to not cut down on the aspect of interchangeability. To reduce the drag force (Equation 1) on articulated trucks the focus has been to lower the shape based coefficient C_D over the whole truck.

One part of a trucks overall C_D value is contributed by flow separation in the boundary layer, which occurs when the kinetic energy of the flow in the boundary layer is not sufficient to overcome the adverse pressure gradient. Figure 1-4 shows this effect in the transition of the airflow from cab roof to trailer roof.

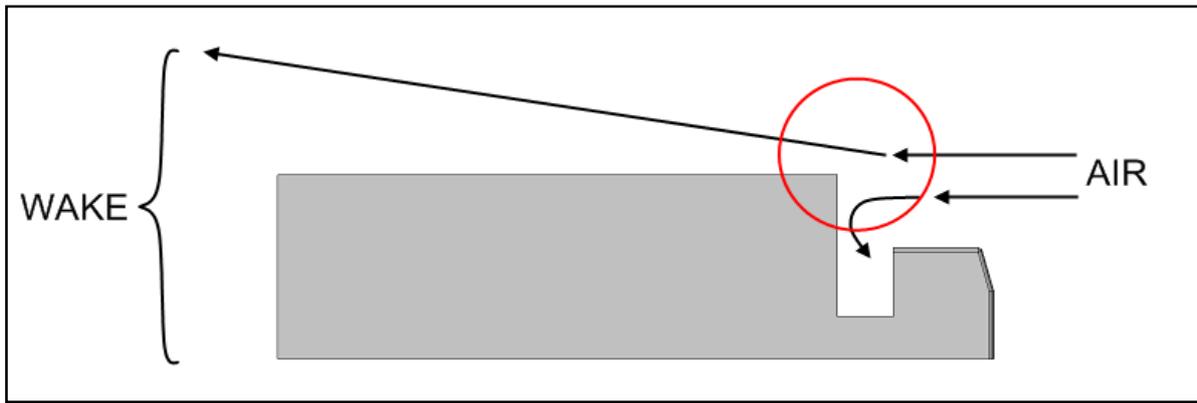


Figure 1-4: Flow separation on an articulated truck (schematic) [6]

To reduce the flow separation and consequently reduce aerodynamic drag on articulated trucks, researchers have been developing aerodynamic add-on devices. These add-on devices can be categorised as two types:

1. Active add-on device

Active add-on devices work by using a secondary energy source to power up. These devices are rarely used in the industry due to high installation and maintenance costs. For example moving surface boundary-layer control (MSBC) devices.

2. Passive add-on devices

Passive add-on devices are used to change the oncoming flow profiles using their shapes. They are widely used add-on devices for truck-trailer units in almost any truck-trailer configurations that currently operate in the UK. Passive add-on devices are much cheaper to manufacture and install compared to active flow control devices. They also have very low maintenance cost. Many different types of passive add-on devices are available. The most common passive add-on devices used for improving the airflow in the area of the truck-trailer transition are listed below:

- i. Cab roof fairing devices
- ii. Corner Vanes
- iii. Side skirts
- iv. Truck-trailer gap seals

Using these devices streamlines the flow transition in the cab-trailer area and hence reduces the flow separation and aerodynamic drag force. This reduces the fuel consumption of the truck and consequently reduces the greenhouse gas emission. Add-on devices for optimizing airflow in the vicinity of a truck-trailer vehicle are available in different styles and follow different concepts. This study focusses on the concept of cab roof deflector devices (CRD).

The aim of CRD parts is to streamline the airflow in front of the cab roof to the trailer roof by deflecting the airflow and to prevent flow separation [7, 8]. Depending on the configuration of deflector size, deflector shape, deflector positioning, and the trailer height the resulting drag force can vary in a range of about $\pm 20\%$. Reversely this means, that an optimized configuration of CRD geometry and CRD positioning for a certain tractor-trailer combination can lower the resulting drag force significantly.

Field experiments have been performed to compare an optimized airflow in the truck-trailer gap area with maladjusted or suboptimal shaped CRDs. The results show that suboptimal CRD configurations lead to a significant increase on overall fuel consumption of articulated trucks [5]. Trailer heights for articulated trucks in the EU are limited by EU guidelines to a maximum of 4.0 m total. In Britain trailer heights are not limited, but it is recommended not to exceed a trailer height of 4.8 m. This recommendation is derived from design guidelines for tunnels and bridges, which are built with an overhead clearance of 5.0 m [2]. While it is not allowed to use British 4.8 m trailers on European roads, it is possible to use European 4.0 m trailers on UK roads. This leads to varying trailer heights with a range of 4.0 m to 4.8 m in the UK.

1.1 Motivation

The greenhouse effect and related energy saving regulations make it necessary to reduce overall energy consumption of HCVs. Overcoming of aerodynamic drag of articulated trucks is a big component of the energy needs and hence overall fuel consumption in the commercial transport sector. It was found that fuel consumed to overcome aerodynamic drag is a huge component of over 50 % in overall fuel consumption for long distance transport journeys. The remaining 50% of overall fuel consumption is split between acceleration and technologies inherit losses such as rolling resistance and friction losses in drive trains.

The drag force (equation 1) is dependent on four factors: the density of the fluid ρ , the vehicle velocity v , the shape based aerodynamic coefficient C_D and the projected frontal area A . The value of the density of air has been considered as a natural constant with 1.225 kg/m^3 . Changes on the vehicles velocity have not been a viable option because of the transport companies needs

to move goods as quickly as possible. Load capacities and HCV gross weight are standardized and have been considered constant. Maximum speed of HCV is regulated by local and European laws at 56 mph (90 km/h) [10]. Lowering the traveling velocities of HCV is not a viable option for the transport companies. Therefore the velocity of HCV has been considered constant but with respect to legal laws. The projected frontal area of HCVs depends on the shape of the used trailer. Both components width and height of the projected frontal area have been developed into a standardized trailer in the transport and automotive companies [3]. Therefore the trailer dimensioning and the resulting projected frontal areas for changing trailer heights respectively have been considered constant. The chosen trailer heights were 4.2 m for European legal trailers and 4.8 m for UK legal trailers. It has been concluded that the only viable option for the reduction of aerodynamic drag on HCV is to reduce the shape based drag coefficient C_D .

Nowadays most tractor cabs have a mounted cab roof deflector to improve aerodynamics in the roof to trailer transition, but these deflectors are often not in optimum position [11]. The reasons for deflectors in not optimum position vary. Older and very cheap CRDs have a fixed fixed shape without a mechanism for adjustment. Slightly more advanced CRDs can be adjusted manually or even automatically, but require the truck to be in standby for adjusting the position. Automated cab roof deflectors that are available as a standard product feature automated adjustment with respect to the trailer height. These adjust themselves with respect to the trailer height, but are not applicable for a wide range of changing trailer heights. They also can not adjust themselves to changing crosswind conditions. Regardless of the actual reason for any particular case the automation of cab roof deflectors can help to improve aerodynamics of vehicles.

Automated cab roof deflectors can be developed in a way, that optimal aerodynamic positioning with respect to trailer height, cab roof height and crosswind condition can be achieved.

1.2 Research Aim

The overarching aim of this research has been to develop an automated cab roof deflector for tractor-trailer combinations with varying trailer heights. The system to be developed should be able to find the trailer height automatically. Also the system should be able to self-position the CRD to an optimum position, depending on trailer height, vehicle velocity and wind condition.

Chapter 2

Literature review

The previous chapter introduced the available add-on devices for streamlining the airflow around a HCV. It highlighted the reduction of the shape based drag coefficient C_D as purchase in HCV aerodynamic optimization. The optimisation of the positioning of cab roof deflectors by development of an automated control system for cab roof deflectors was defined as research aim.

This chapter reviews the research on HCV aerodynamics that focussed on the reduction of the C_D value. The review begins in the year 1985 with the results on aerodynamic optimisation on HCV after the 2nd oil crisis of 1979. The literature review ends with the latest insights in development of CRD for HCVs and the definition of specific research objectives for this project.

2. Literature review

2.1 Aerodynamics of articulated trucks

Development of trucks and other vehicles has reached a level of efficiency for any type of engine and drive train component, so that big improvements on the state of the art engineering in this field are hard to achieve. Starting in the 1970s, scientists and engineers all over the world have been working to improve aerodynamics feature of trucks, cars and other vehicles. Researchers have been developing aerodynamic enhancement components with very different utilities. In the early years of aerodynamic research scientists have been utilizing wind tunnels and simple shapes. Since the year 2000 researchers have been utilizing refined computational models of aerodynamic streamlining for any type of vehicles. With technological progress the shapes of additional aerodynamic components have been improved to high complexity structures. In reference to trucks and trailers this means that all aerodynamic expertise gathered over the years has been applied for truck trailer combinations. This has been leading to significant improvements in aerodynamic drag reduction and fuel consumption reduction. Nevertheless aerodynamic optimization is a meaningful field to automotive engineers.

Truck aerodynamics has undergone significant improvements since then. Flat cab roof deflectors were successively replaced with individually shaped glass fibre based free form deflectors. Some of the latest cab roof deflectors are made adjustable in height to optimize aerodynamics of the vehicle according to the interchangeable trailers used. These height adjustment constructions are adjusted manually by the driver or remotely adjusted with a cable bound remote control. The most refined semi automated CRD systems have a sensor for detecting a certain position in reference to a trailer and signalling when a predefined position is reached. But these semi automated deflectors still need an operator. These semi automated CRD systems are remotely controllable, but can not be operated during a trucks journey and cannot respond to changing wind conditions. Cross wind conditions however significantly change aerodynamic drag of vehicles during journeys [7]. Aerodynamic drag in crosswind conditions can be improved by adjusting wind deflector components of trucks in real time to changing wind conditions.

As discussed earlier aerodynamic drag is the main reason for fuel consumption on HCVs. Up to 65 % of overall fuel consumption can be related to aerodynamic drag (see Figure 1-2). Since design and construction of trucks and trailers is standardized by local and European laws most

research projects address drag reduction aims with aerodynamic enhancement parts such as CRD. CRD are used to optimize the airflow transition from the cab roof to the trailer roof.

A big part of aerodynamic drag is developed from flow separation with a high pressure on the front surface of the vehicle and a low pressure wake behind the rear surface of the vehicle. The combination of both low and high pressure fields results in pressure drag. First experiments and studies in the area of aerodynamics were performed as early as 1953, but did not have any impact in the transport industry. Vehicle aerodynamics first became a concern when the oil and fuel prices multiplied in the years of 1972 to 1981. The first oil crisis raised the awareness in the transportation companies to reduce fuel cost of their transport fleets.

Garry [8] presented review of aerodynamic developments on trucks up to the year 1985. He explained that CRD were developed in the 1960s with the purpose to streamline the aerodynamics on trucks. Despite the aerodynamics engineers commitment on explaining how aerodynamics work CRD did not become widely accepted. Because of a high variety in trucks and trailers and especially their combination the industry developed many proprietary CRD. When the oil price multiplied after the 1973 crisis and did not go down again companies started to adapt CRD as default equipment for trucks. In his summary Gary points out, that just mounting basic aerodynamic parts on trucks is not enough. He states that it is important to match the cab body and the trailer body to obtain uniform flow field.

Figure 2-1 shows a number of top view schematics with different cab and trailer combinations. With combination of different shapes and sizes of the cabs the interaction of body shapes and creation of swirls in airflow is visualized. It can be expected that a matched flow field results in less aerodynamic drag than a mismatched flow field. In the third schematic of Figure 2-1 is shown that a trailer and truck with same width and rounded vertical corners results in a matched flow field. This knowledge was already adapted to truck development in the year 1985, but is not practical when cab-trailer gaps are too long.

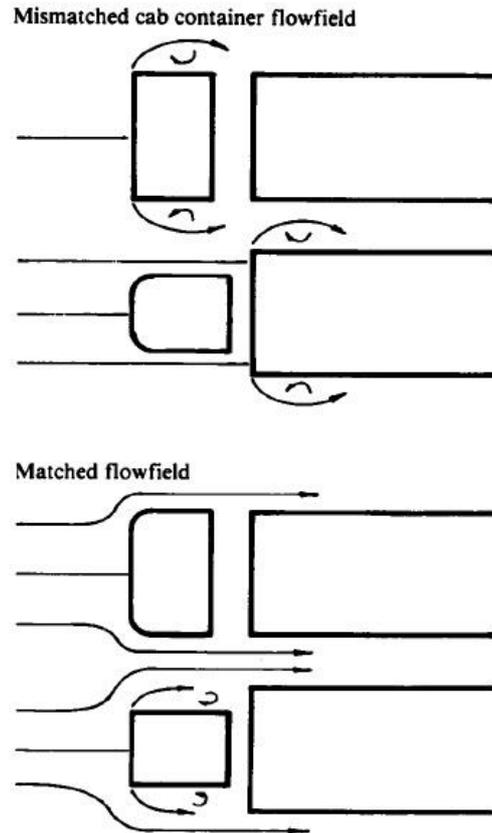


Figure 2-1: Matching the cab and container flow field for optimum aerodynamic interference [8]

Trailer fore body fairings could be used to reduce the gap drag in cases of aerodynamically suboptimal gap lengths. Trailer fore body fairings reduce flow separations and create an overall effect of streamlining the airflow around a vehicle. This effect reduces up to 40% aerodynamic drag in zero yaw wind condition. As an alternative to fore body fairings trailer edge fairings are mentioned. These have a lesser drag reduction potential of 27 % but are easier to apply because of their overall smaller dimensions. Gary states that in yaw angle wind condition these aerodynamic enhancements do not have much of an impact. This is because the pressure difference in the windward side and the lee side causes a cross stream through the cab-trailer gap. That cross stream leads to a flow separation at the vertical trailer edges leading to increased drag in cross wind situations. He concludes that the most efficient aerodynamic devices articulated trucks are a combination of CRD and matched trailers or fore body fairings for trailers respectively. Applying these components a reduction of aerodynamic drag of about 30 % can be expected. 30% reduction of aerodynamic drag correlates with a reduction in fuel consumption of about 8 % to 12 %. These results vary depending on drive cycles, vehicle load

and vehicle type. With consideration of cross wind conditions the development of trailer front fairings is suggested to be more effective.

Drollinger [7] summarized aerodynamics knowledge regarding truck aerodynamics in his publication. He explains distribution of fuel consumption of trucks to rolling resistance, pressure on the front and back surfaces respectively, the tractor-trailer gap effects, drag from under body construction and skin friction effects. The author compares the aerodynamic drag reduction achieved by a number of different aerodynamic add on devices. He states that the aerodynamic effect of CRD devices lead to a drag force reduction of about 24 %. Combination of aerodynamic add on devices can increase the drag reduction effect. These results are about the same as given by Gary before [8]. Drollinger concludes with an assumption that it is likely to find even more effective aerodynamic enhancement components with new kinds of devices that were not developed yet. The difficulty for development of new aerodynamic enhancement of HCV is that new truck and trailer concepts need to be accepted by the transport companies. This makes it hard to develop a completely new truck design that focuses on aerodynamic streamlining. Nevertheless it is necessary to improve the design of HCV with respect to aerodynamics. Performance tests of aerodynamic enhancement parts can be performed with any known analysis technology as all technologies have characteristic (dis-)advantages.

Figure 2-2 shows results of wind tunnel experiments performed with class 8 tractors and cab-over-engine tractors (COE). The resulting drag coefficients for a stock and aerodynamically optimized truck were normalized against the drag coefficient of a conventional truck with a 32 inch gap. It can be seen that for both COE trucks and conventional trucks the gap has an impact on aerodynamic drag and is increasing gradually. For wind tunnel testing of devices Drollinger points out, that even small scale wind tunnels give good results if the Reynolds numbers of the model are identical with the full scale truck.

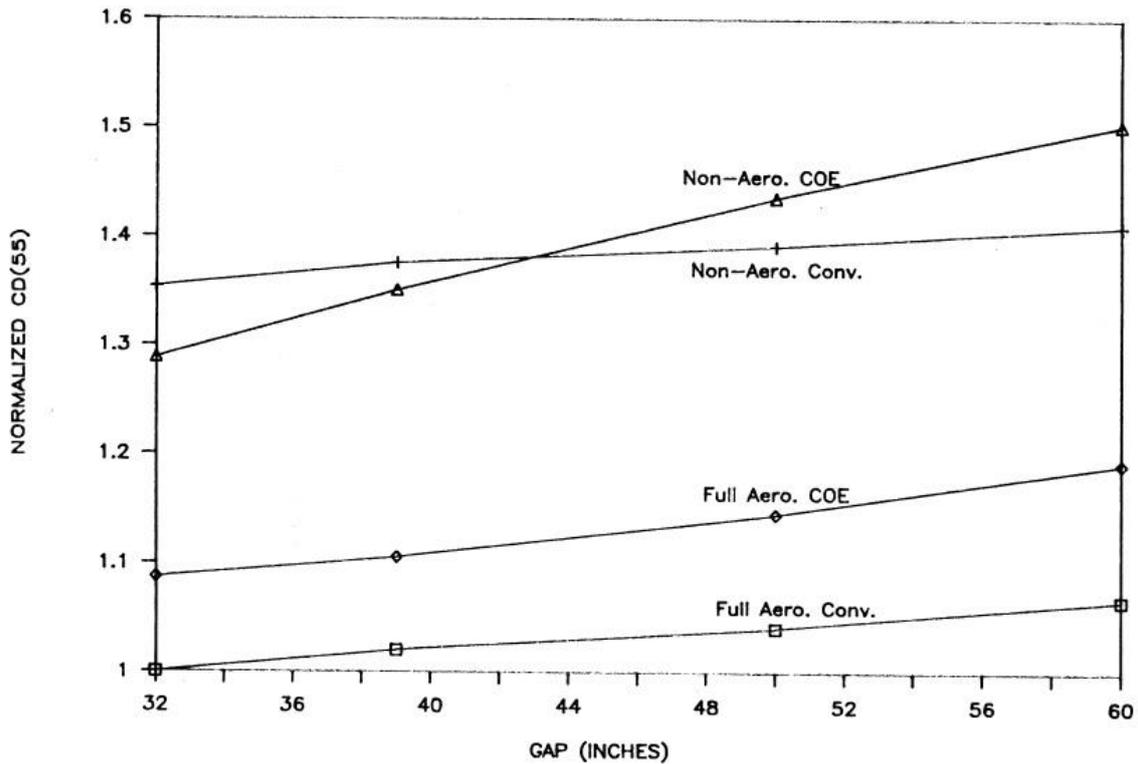


Figure 2-2: Example curves of normalized drag coefficient as a function of tractor/trailer gap for several vehicle configurations [7]

Modi et al. [12] performed a set of 1:6 scale wind tunnel experiments on cube-vans. He studied the effect of boundary layer control by fences mounted on the front size of the load space. Modi concludes that mounting fences on the front of the load space box can reduce the aerodynamic drag by about 9.6 % for steady movement of about 100 km/h. However this research results have proven usability it was not applied to rigid trucks, probably because of a lack in acceptance in the transport industry.

Since then several other research results were adapted into heavy duty vehicle development, leading to the regulations and standardisations for semi-trailer mentioned before. Aerodynamic improvements of cab and trailer styling were adapted. This included rounded vehicle edges, reduced cab-trailer with standardised kingpin positioning and reduced gap length. Furthermore aerodynamic add on devices such as fixed cab side extensions and CRDs became standard equipment. The stream separation in the truck-trailer gap area had been understood and consequently was improved. The awareness on aerodynamic drag and the increased acceptance of the aerodynamic enhancements lead to a significant improvement on fuel consumption and greenhouse gas emissions.

Dominy [11] focused his work on the effectiveness of cab roof deflectors. His work focused on optimizing the CRD position and evaluated drag forces on trucks with CRD on journeys with and without attached trailers. Therefore a set of wind tunnel measurements was performed for streamlined trucks with and without trailers. The data for a streamlined truck with trailer was used as reference to validate the other dataset for a streamlined truck without trailer. Based on these results a software model was developed. The model calculates reasonable fuel consumption data for a predefined route based on information about the truck model. With that software model he studied the influence of different cab roof deflector shapes on aerodynamic drag and fuel consumption. Also the impact of misaligned CRD was researched. Dominy concluded, that cab roof deflectors only have a positive effect on drag and fuel consumption if they are adjusted correctly to the trailer height. He found that a correctly positioned cab roof deflector can reduce fuel consumption by 4.6 % over the same vehicle without cab roof deflector. If the CRD not lowered in case of a truck operating without trailer, this in fact increases the fuel consumption by about 3.5 % over the same vehicle without CRD. These results match with the drag equation (equation 1). Increasing the frontal projected surface area increases the resulting dragforce on the vehicle. Therefore the mode of operation, with trailer or without trailer, should be considered.

Dominy's [11] results show that correctly adjusted cab roof deflectors, depending on the actual deflector shape, can reduce fuel consumption for journeys with trailers by about 2 litres per 100 km. On the flipside the same deflectors that reduced the fuel consumption when operated with an articulated truck increase the trucks fuel consumption when operated without trailer. The additional fuel consumption is between 1.5 litres per 100 km and 4.5 litres per 100 km, depending on the deflector shape. It can be concluded that it is important to adjust the cab roof deflectors to the mode of operation for maximum fuel saving.

McCallen et al. [9] used 1:14 scaled wind tunnel experiments to validate first available CFD methods. Both CFD and wind tunnel experiments were used to research the correlation of truck-trailer gap length and the vehicle dimension. The truck model featured a fully streamlined cab-over-engine design. The drag was measured at a wind speed of 22 m/s, which is about 80 km/h or 50 mph. The drag coefficient was related to G/L , with G as gap length and L is the square root of the frontal area. They found out, that the aerodynamic drag increases gradually with increasing G/L ratio. These results match the results as found earlier by Drollinger [7] and prove the used CFD model as performing good. Extending the experiments over a wider range

of gap dimensions brought up a new result: When the G/L ratio exceeds a value of 0.5 the drag force significantly increases over the linear gain, representing a critical value for HCV aerodynamics. The related gap length at the mentioned critical value can be calculated from the cited value of $L = 0.218$ m in 1:14 scale, which equals to a gap length of $G = 1.5$ m in full scale. It can be concluded that truck-trailer gap lengths smaller than 1.5 m are worthwhile.

Wood et al. [13] focussed their work on turbulence and vortex modifying devices to reduce aerodynamic drag and increase fuel efficiency of articulated trucks. Their goal was to develop easy to maintain add-on devices depending on available knowledge about stream behaviour in the cab-trailer gap area and the rear end of a trailer. To improve the vortex related drag the authors developed devices named cross-flow vortex trap device, vortex strake device and undercarriage flow device. The vortex strake device and the undercarriage flow device are mounted beneath the trailer. They are designed to shape the trailer rear vortices in an aerodynamically advantageous way.

The cross-flow vortex traps are seven vertical surfaces mounted on a trailer front. Those create controlled vortices in crosswind condition. The vortices created by the vortex traps result in an aligned force on the crosswind airflow entering the gap. The aligned flow created by the vortex traps lowers the crosswind related drag in the gap area (see Figure 2-3).

Evaluation of the effectiveness of the devices has taken place with experimental devices mounted on a truck, comparing effectiveness in overall fuel consumption reduction over a number of journeys. Based on their evaluation datasets the authors conclude that the use of their cross-flow vortex trap device can improve fuel economy by 3.5 to 8 %.

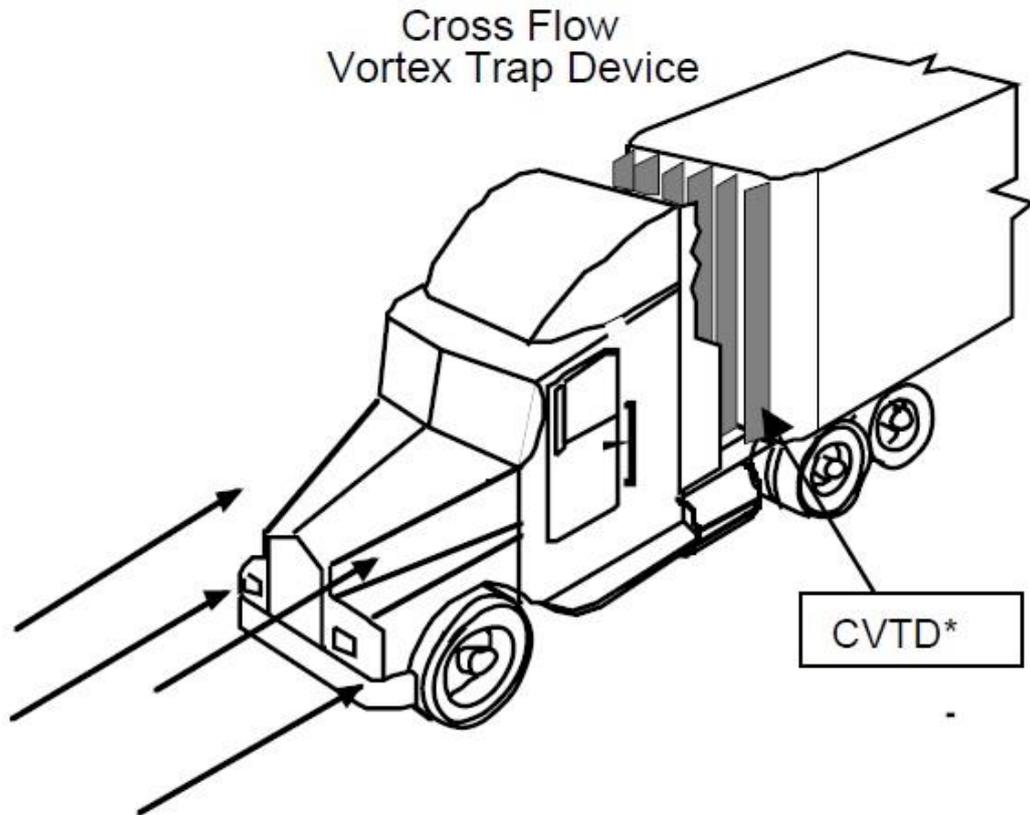


Figure 2-3: Sketch of the cross flow vortex trap gap treatment device installed on the trailer front face. [13]

Cooper et al. [14] performed a wind tunnel study for evaluating recent insights in aerodynamic styling of semi-trailer trucks. The study has been performed with a 1:10 scale model in a small wind tunnel and a full scale truck in a 9x9 m wind tunnel. The aerodynamic enhancements included cab mounted side extenders, trailer mounted gap sealing plates, trailer side skirts, trailer mounted angled boat-tail plates and different under body fairings. The experiments were run with a basic truck for first. In the later experiments aerodynamic enhancements were successively added to the articulated truck. Wind conditions were set to 55mph and yaw angle was 0 ° to 10 ° with increments of 2 °. The performance for all tested add-on devices was measured. Also combination of add-on devices were tested for the best overall performance. Cooper et al. conclude that with the adaption of the best combination of all known technologies of drag reduction on articulated trucks aerodynamic drag coefficient C_D can be reduced from 0.716 to 0.580.

The department of transport of the United Kingdom [15] performed a project, where they investigated the effect of using a streamlined trailer with lowered front edge on fuel consumption. This study was performed in cooperation with the company of Somerfield. Half of the Somerfield truck fleet was equipped with the modified trailers and the other half of the fleet remained unmodified. After one year of everyday usage the fuel consumption of the two fleet parts were evaluated and compared against each other. Obtained results show that only the lowered trailer front edge and the resulting aerodynamic streamlining had an impact of about 7% reduction in fuel consumption per year.

H. Chowdhury et al. [16] analysed the impact of combinations of trailer side skirts, front fairings and gap covering parts against a stock Australian articulated truck in a 1:10 wind tunnel study. Their set of experiments included various combinations of yaw angle from 0 ° to 15 ° in 5 ° increments, wind speeds from 40 km/h to 120 km/h in 10 km/h increments and various aerodynamic enhancement parts. The comparison parameter for the effectiveness of the aerodynamic enhancements was aerodynamic drag, which was measured with a multi-axis load cell. The authors compared the drag coefficient C_D for different combinations of trailer side fairings with combinations of cab roof deflectors.

Figure 2-4 shows the different combinations of aerodynamic enhancement components that were evaluated. From this test the effectivity of streamlining in the truck-trailer gap section can be seen. To do so the drag coefficients with the same add-on setup, but different gap fairing configurations were compared. Configurations a-b, c-e and d-f only differ in the gap fairing configuration and are identical in configuration of other devices.

Figure 2-5 shows the shape based drag coefficient C_D as a function of changing velocities for each of the configurations. Comparison of the drag coefficients of configurations a-b, c-e, d-f shows the effect of perfect streamlining in the gap area. The results show that an average drop in drag coefficient C_D of 10% or 0.06 is related to just streamlining the gap area.

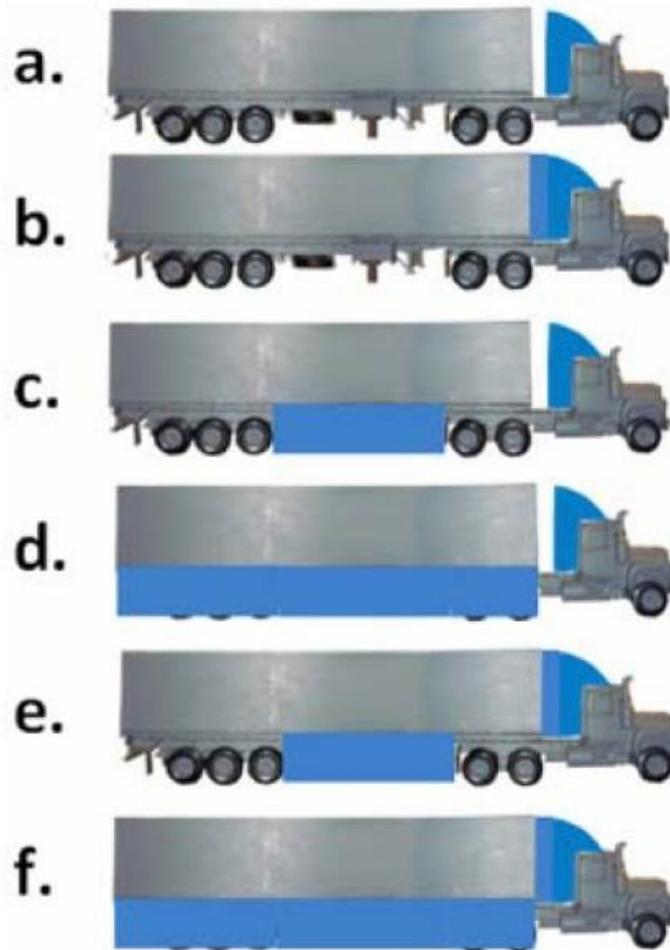


Figure 2-4: Different combinations of fairing on the baseline semi-trailer truck model [16]

It can be concluded, that aerodynamic drag can be reduced by streamlining the airflow over the unavoidable gap between cab and trailer. A cab roof deflector that is able to respond to crosswind conditions can help reducing drag of articulated trucks.

Several studies have been performed with the focus to find out how aerodynamics of trucks and truck trailer combinations can be improved. Many results have made their way into standard styling of trucks, trailers and additional aerodynamic equipment. Size in length, width and height of truck trailer combinations became strictly regulated by European and local laws [2]. Because of this standardisation the shape of commercial vehicles has not changed much in the last decade. This led to more focussed research projects on additional aerodynamic parts for commercial vehicles than on shape of vehicles or trailers themselves.

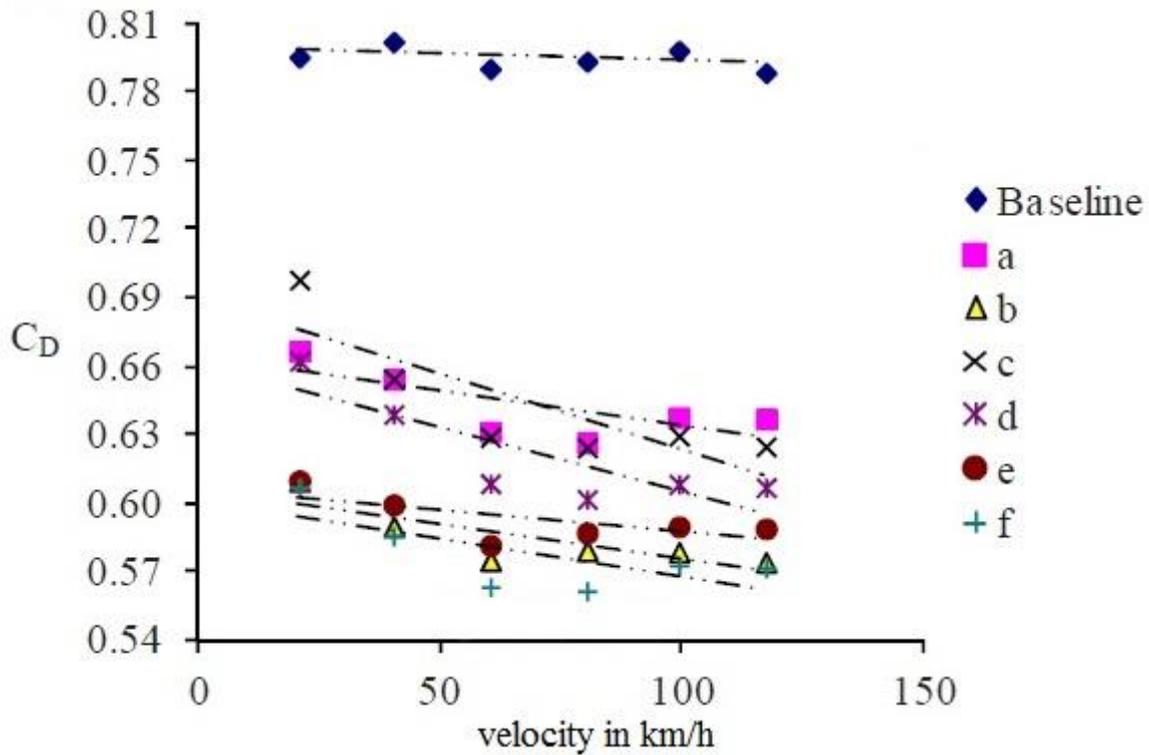


Figure 2-5: Drag coefficient as a function of speed for different test configurations at yaw angle $\beta = 0^\circ$ [16]

Automotive industries have realized the need of further development on aerodynamic enhancements in the gap area of articulated trucks. They started the development of automatically adjustable devices and the required sensor arrays.

Hatcher components ltd. [17] in cooperation with Cranfield university and Daimler AG developed an automated cab roof deflector system. The system uses surface mounted pressure sensors on the cab roof deflector and input from the on-board computer. In this concept information about the trailer height is acquired electronically via the board-net of truck and trailer. Alternatively the trailer height information can be entered manually. The vehicle speed for calculation of the CRD position is taken from the on board speedometer. If all components are connected correctly the deflector adjusts itself to changing wind conditions during the vehicles trip. The patent covers the Hatcher Active Freddie™ product [18] that was released in 2013. Since the system is highly integrated with the truck and trailer board net it is very specific in its requirements for the configuration and brand of the articulated truck.

Mercedes [19] also registered a patent in the USA that describes wind condition sensing on the trucks surface by surface mounting pressure sensors directly on the truck and trailer surfaces.

From the pressure information the flow profile around the articulated truck is calculated. The control system tries to minimize turbulent conditions by adjusting movable wind deflector components. Minimizing turbulent flow on trailer surface areas leads to more streamlined airflow over and besides the articulated truck and equally reduces aerodynamic drag. Because the sensor system is distributed among truck and trailer this requires a highly integrated system too.

Scania CV AB [20] registered an international patent that describes a 3 piece wind deflector setup with two side fairings and cab roof deflector. These aerodynamic parts are connected to a motor with a built in sensor and a control unit. That built in sensor measures the required current to keep the attached deflector in the desired position. The pressure applied to the add-on devices is calculated from the measured current. This information is converted into the applied pressure and related wind condition. Several modes that follow different possible strategies for drag reduction are available. These strategies are based on different parameters that are obtained by the implemented sensors.

Discussed studies show the research results in the field of aerodynamics that have been made in the past three decades. They also show the improvements in aerodynamic streamlining of articulated trucks that were developed. Several studies have been carried out to find the optimum combination of aerodynamic enhancement components for HCVs. It was shown that the airflow behaviour in the cab-trailer gap area has a major impact on overall drag [8, 9] and is directly related on fuel consumption of the vehicle [4, 7,]. A field study with focus on fuel consumption reduction has shown that only streamlining the cab roof deflector to trailer roof transition can reduce the overall fuel consumption by 7 % [15]. Although it is known that airflow optimisation in the cab-trailer gap area is not yet used to the fullest [11] research groups focussed on development and evaluation of new kinds of aerodynamic devices [12, 13, 14].

Though it was not the focus of these studies it can be observed, that cross wind conditions change the aerodynamic characteristics and the magnitude of the resulting drag force [7, 8, 14]. It was shown, that a perfectly positioned cab roof deflector can create a 4.6 % reduction in fuel consumption compared to the similar vehicle without any cab roof deflector mounted [11].

Automotive companies such as Mercedes and Scania have realized the potential of aerodynamic optimizations in automated airflow guiding devices. Though at least two companies are working on automated airflow guiding devices only one patent has made its

appearance as a product yet [17, 18]. Hatchers Active Freddie™ product is able to sense the wind condition on the HCV and adjust itself to changing wind conditions. The system can adjust itself to changing wind conditions but it is limited to specific trailers with a certain height. It is not compatible with varying trailer heights and not compatible with different brands of trucks and trailers.

2.2 Specific research objectives

The literature research has brought up, that the potential in aerodynamic streamlining in the cab-trailer gap area is underused. It can be improved by automating CRD and gap reducing cab side fairings that automatically adjust to changing wind conditions. Automotive corporations have realized the potential of automated aerodynamic enhancement components. They have been working on different concepts for wind condition measuring sensor arrays. The sensors arrays adapt different concepts in terms of used sensors and how sensors are utilized.

Mercedes' first concept is to measure the magnitude of turbulences in the cab-trailer gap area with surface mounted pressure sensors on the cab rear surface and the trailer front surface. The obtained sensor values are used to minimize turbulences in the gap area and consequently reduce aerodynamic drag [19]. Mercedes' second concept uses surface mounted pressure sensors to measure the pressure on the surface of a cab roof deflector shield [17]. The sensor readings obtained from the CRD surface are converted into wind information. The wind information is used to adjust the CRD for drag reduction. Scania [20] uses current measuring sensors which are implemented with the actuators for the adjustable components. These sensors measure the current that is needed to keep the aerodynamic components in position and derive wind condition from these sensor readings.

In order to develop a control system for automated cab roof deflectors this research follows a different approach in how sensors are used. The approach is to directly measure the wind condition and not working with surface pressures obtained by pressure sensors mounted on vehicle surfaces.

The objectives for this research project of developing a control system for automating CRD are as follows:

- To implement multi-hole probe technology to determine the vehicle speed and the crosswind condition. This involves the development and calibration of multi-hole probe, which will be used to measure the magnitude and direction of the wind vector.

- To develop a control system, that is applicable for different trailer heights and three dimensional wind conditions. This involves development of a sensor array for the detection of different trailer heights and the development of a microcontroller based control circuit for the automated control system.
- To develop a smart software for the control system that supports the measurement of the wind vector, the detection of the trailer height and a smart algorithm for the automated optimum positioning of the cab roof deflector. The software should be easily adaptable.

Chapter 3

Development of a multisensory probe system for quantification of wind angle in the vicinity of a vehicle

The previous chapter presented the development of research on HCV aerodynamics over the last three decades. The research results in the field of aerodynamic add-on components for HCVs has been presented. The review of the research results brought up that the effectiveness of air guiding devices can be improved by automation of these devices. Automated air guiding devices can adjust themselves to changing wind conditions that are encountered on the journeys of HCVs and consequently reduce fuel consumption

This chapter sheds light on the various types of pressure probes for sensing wind information. It discusses the development and calibration of multi-hole probe to quantify wind velocity vector. It also presents a calibration method for a multi hole probe and how the calibration parameters are implemented with a multi hole probe wind sensor system.

3. Development of a multisensory probe system for quantification of wind angle in the vicinity of a vehicle

3.1 Pitot Tube

Pitot tubes are common and well known probes for measuring fluid velocities in a variety of applications, such as avionics, wind velocity measurement and formula one racing cars. Pitot tubes usually are metal rods which are bend rectangular (other shapes are also known, but less applicable). They have two holes, of which one is facing the fluid flow to be measured and the other one is orientated orthogonally to the fluid flow (see Figure 3-1).

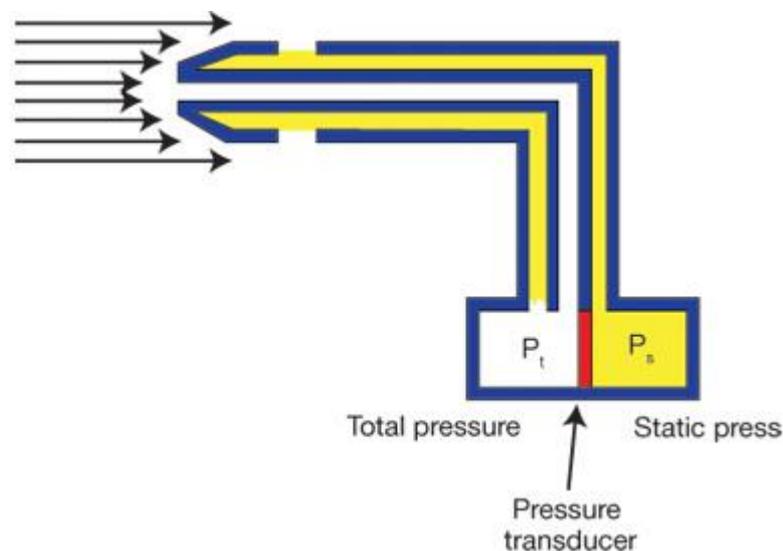


Figure 3-1: Principle of a pitot tube probe [21]

The fluid in the tube that is facing the flow stagnates and builds up a pressure, which is known as stagnation pressure or total pressure. The hole orthogonally to the fluid stream measures the pressure in the flow, which is known as static pressure. The Bernoulli principle states, that

$$P_{total} - P_{static} = \frac{1}{2} \rho v^2 \quad (\text{Equation 2})$$

where P_{total} as total pressure, P_{static} as static pressure, ρ as density of the fluid and v the velocity of the flow. That equation can be solved for velocity v and can calculate the flow velocity with the measured pressures, when the fluid density is known:

$$v = \sqrt{\frac{2*(P_{total}-P_{static})}{\rho}} \quad (\text{Equation 3})$$

As stated before, pitot tubes can only be used for measuring flow velocity, but cannot measure the direction of the flow. Measuring flow velocity and flow direction requires different probes with more pressure values measured simultaneously.

3.2 Three-hole probe

Form the above discussions, it can be noticed that two-hole probe or pitot tube can only measure wind velocity. For the measurement of flow velocity and flow direction, probes with capability of simultaneously measuring more than P_{total} and P_{static} are required. Three hole probes are two dimensional probes. They are composed of 3 aligned pressure sensing tubes. These probes can measure the velocity of a flow as well as a flow angle in a plane. Three hole probes have to be calibrated for precise measurement of velocity and angle. A schematic of a three hole probe tip is shown in figure 3-2.

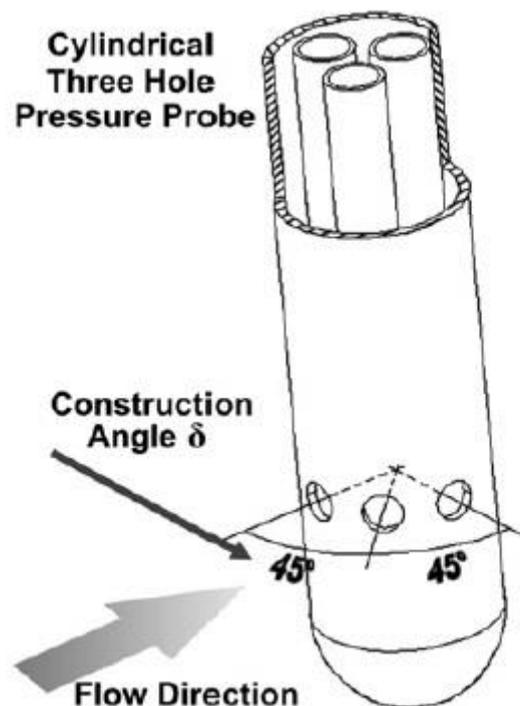


Figure 3-2: Schematic of a cylindrical two dimensional probe tip [22]

The calibration of this type of probes is required because the pressure distribution on the pressure holes is defined by the shape of the probe tip and the distribution of the holes on the probes surface. Calibration of multi hole probes have a common feature, which is the use of normalized pressure coefficients. These coefficients for a three hole probe are obtained as a function of total pressure P_{total} , the static pressure P_{static} , the pressure on the centre hole

P_{center} , as well as the pressure on the left hand hole P_{left} and on the right hand side P_{right} resulting in the following equations:

$$C_{\alpha} = \frac{P_{left} - P_{right}}{Q} \quad C_{P_T} = \frac{P_{total} - P_{center}}{Q} \quad C_{P_S} = \frac{P_{total} - P_{static}}{Q} \quad (\text{Equation 4})$$

with

$$Q = P_{center} - \frac{1}{2}(P_{left} - P_{right}) \quad (\text{Equation 5})$$

where c_{α} is the yaw angle coefficient, $c_{P_{total}}$ and $c_{P_{static}}$ are the coefficients for total and static pressure. Q is the pressure normalization factor obtained from the probes pressure readings. While the flow angle c_{α} can directly be calculated from the pressure readings of the two left and right hole pressures, the probe needs to be calibrated for the coefficients $c_{P_{total}}$ and $c_{P_{static}}$. Calibration for these two coefficients allows to precisely calculate total pressure, static pressure and wind velocity respectively [22].

Applications that require the measurement of three dimensional flow condition cannot utilize three hole probes because of their limitation in only measuring the flow angle in one plane. Combining two 3 hole probes that are rotated 90° against each other allows to measure the flow field in two planes that are oriented vertically against each other. Measuring velocity and two independent flow angles gives full three dimensional flow field information. That principle was adapted to develop five hole probes. Five hole probes have become well known sensors for three dimensional flow field measurement.

3.3 Five-hole probes

Five hole probes are common and well known probes of the group of multi hole probes. They are widely used in various applications such as aeronautics and sensor applications for e.g. invasive flow measurement in pipes and pipelines. Five hole probes cannot only sense the dynamic pressure in a flow and therefore obtain information about the flows velocity, but can also sense information about direction of the flow. Five hole probes can measure full three dimensional flow information, such as pitch angles, yaw angles and velocity [23].

As mentioned previously the measurement of flow condition is always an indirect measurement that requires good knowledge about physical relations to set up a viable flow measurement experiment or application. An important aspect of pressure probes is that the probe outputs the

same signals when the probe is moving in a steady flow field or when the probe is steady in a moving flow field. Examples for one of each application are a pitot tube mounted on a formula one racing car for measuring the velocity of the car (moving probe) and a pitot tube mounted in a pipeline for measuring the velocity of the flow stream (moving flow field).

Drollinger [7] pointed out, that crosswind conditions are not negligible in considerations on aerodynamics and drag improvement of articulated trucks. He states that the wind relative to the vehicle is the vector sum V , of the crosswind vector V_w and the truck velocity vector V_T , as shown in figure 3-3.

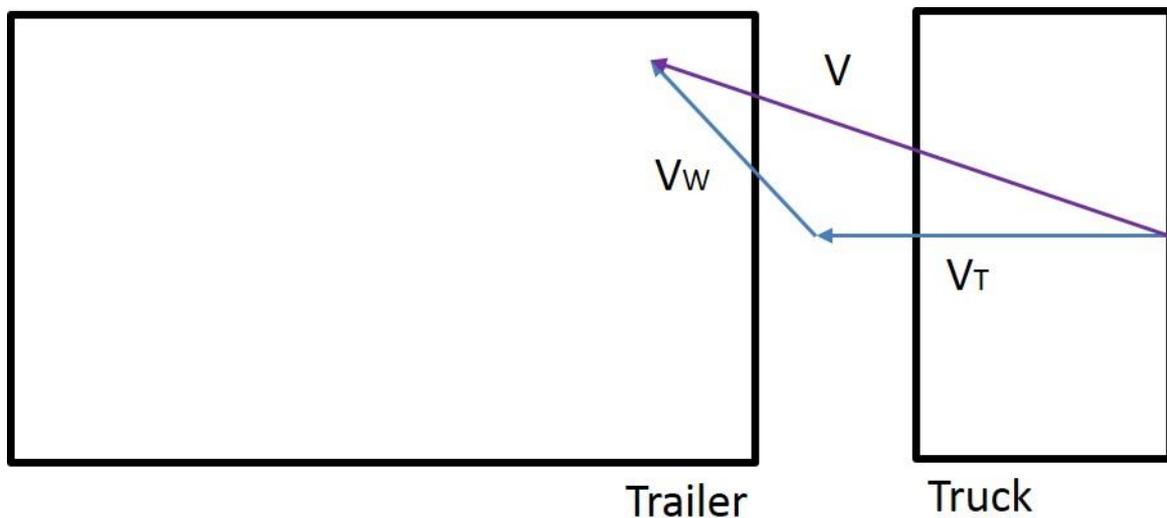


Figure 3-3: Wind vectors on a truck in crosswind condition [7]

The vector wise addition of truck velocity and crosswind shifts the wind vector relative to the truck V when the truck is driving on an incline (Fig. 3-3). Vector wise addition of flow velocities in the environment of a driving vehicle also represents, that crosswinds have an impact on the magnitude of the resulting three dimensional flow vector V . This leads to the conclusion that wind condition measurement cannot be replaced with vehicle velocity fed into an aerodynamic control system from an external source, for example from the trucks speedometer.

Getting precise wind information including three dimensional cross wind components is the basis for the aim of developing a system that adjusts itself to aerodynamic optimum position. The correct positioning of the cab roof deflector is dependent on the cross wind situation on a trucks journey. Hence a five hole probe was chosen for the measuring of the wind vector for

this application. Five hole probes always feature slight mechanical diversities within the manufacturing tolerances and therefore need to be calibrated. The calibration of a probe compensates the influence of the mechanical diversities. Only with a calibrated probe and calibration parameters implemented with the probe system, precise wind information can be obtained. The calibration parameters are unique for each probe and need to be implemented with the control system that is connected to the probe.

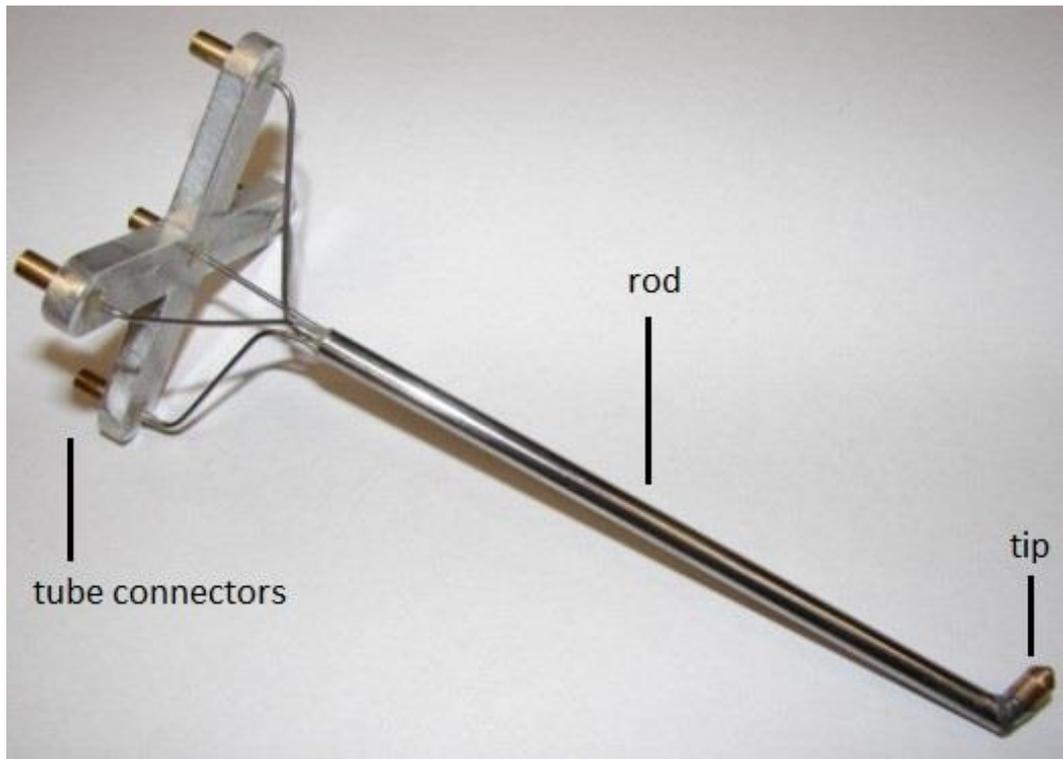


Figure 3-4: Labelled picture of the used five hole probe

Five hole probe sensor systems consist of the probe tip, probe rod (Figure 3-4), tubes connecting the five holes of the probe with the five pressure sensors and a micro controller that evaluates the sensor outputs (Figure 3-5). The sensor outputs are processed to actual wind information of pitch angle, yaw angle and wind speed. The nomenclature for five hole probes follows the nomenclature that was introduced by Krause and Dudzinski [24]. The centre hole is labelled as number 5. The remaining four holes are labelled clockwise as number 1-4, starting with the top hole as number 1. The pressures obtained by the connected pressure sensors are called P_1 to P_5 with respect to the connected probe hole (Figure 3-5). The pitch angle is defined as angle alpha and the yaw angle is defined as angle beta. The probes aerodynamic centre is when the alpha and beta angles of the stream are equal to zero and the stream flow is directed exactly towards the centre hole.

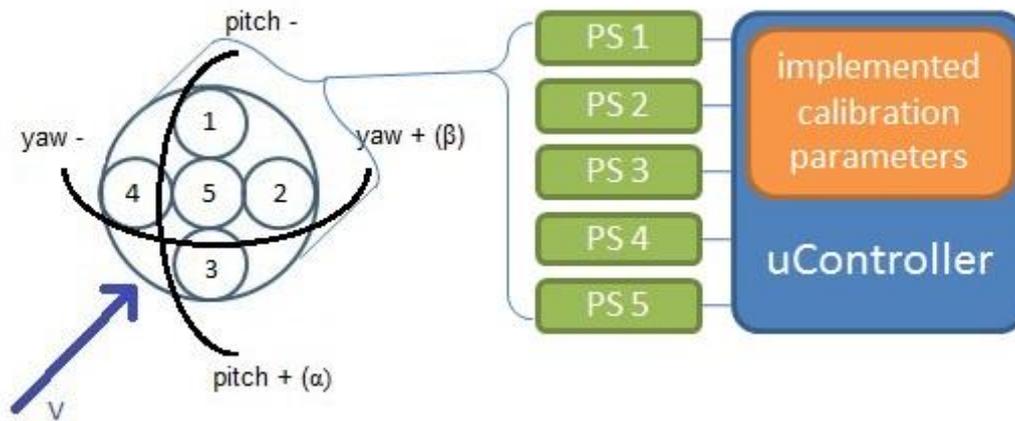


Figure 3-5: Nomenclature of a five hole probe and setup of a five hole probe sensor system

3.4 Selection of pressure sensors

To match system requirements and sensor characteristics an estimation of the pressure range has been made. The expected pressures on the probe tip have been calculated from the wind speeds in real application, including velocities of trucks on motorways as well as common wind velocities. The speed limit for trucks in the UK is 56 mph [10]. According to the Beaufort wind force scale (Figure 3-6) the lowest force of dangerous wind velocities is above 24 mph. The maximum relative wind velocity has been considered as vector wise addition of truck velocity and wind velocity (Figure 3-3). The expected maximum wind velocity relative to the truck is about 80 mph (when the truck drives directly against the wind breeze). A wind velocity of 80 mph equals 35.5 m/s or a dynamic pressure of 772 Pa (equation 3).

**BEAUFORT WIND FORCE SCALE:
Specifications and equivalent speeds for use at sea**

FORCE	Equivalent miles/hr	Speed knots	Wave m	Height ft	Description	Map Symbols	U.S. Advisory Flags	SPECIFICATIONS FOR USE AT SEA
0	0-1	0-1	0	0	Calm			Sea like a mirror
1	1-3	1-3	.1	.33	Light Air			Ripples with the appearance of scales are formed, but without foam crests.
2	4-7	4-6	.2	.66	Light Breeze			Small wavelets, still short, but more pronounced. Crests have a glassy appearance and do not break.
3	8-12	7-10	.6	2	Gentle Breeze			Large wavelets. Crests begin to break. Foam of glassy appearance. Perhaps scattered white horses.
4	13-18	11-16	1	3.3	Moderate Breeze			Small waves, becoming larger, fairly frequent white horses.
5	19-24	17-21	2	6.6	Fresh Breeze			Moderate waves, taking a more pronounced long form; many white horses are formed. Chance of some spray.
6	25-31	22-27	3	9.9	Strong Breeze			Large waves begin to form; the white foam crests are more extensive everywhere. Probably some spray.
7	32-38	28-33	4	13	Near Gale			Sea heaps up and white foam from breaking waves begins to be blown in streaks along the direction of the wind.
8	39-46	34-40	5.5	18	Gale			Moderately high waves of greater length; edges of crests begin to break into spindrift. The foam is blown in well-marked streaks along the direction of the wind.
9	47-54	41-47	7	23	Severe Gale			High waves. Dense streaks of foam along the direction of the wind. Crests of waves begin to topple, tumble and roll over. Spray may affect visibility.
10	55-63	48-55	9	30	Storm			Very high waves with long over-hanging crests. The resulting foam, in great patches, is blown in dense white streaks along the direction of the wind. On the whole the surface of the sea takes on a white appearance. The 'tumbling' of the sea becomes heavy and shock-like. Visibility affected.
11	64-72	56-63	11.5	38	Violent Storm			Exceptionally high waves (small and medium-size ships might be for a time lost to view behind the waves). The sea is completely covered with long white patches of foam lying along the direction of the wind. Everywhere the edges of the wave crests are blown into froth. Visibility affected.
12	73-83	64-71	14+	46+	Hurricane			The air is filled with foam and spray. Sea completely white with driving spray; visibility very seriously affected.

Figure 3-6: Beaufort scale for wind velocities

The actual sensor type for converting pressures on the five hole probes holes into electric signals in the laboratory setup have been chosen to be five pressure differential sensors of the type Freescale Semiconductor MPXV7002 [25]. That sensor type features an output signal of 0.5 V DC to 4.5 V DC with a resolution of 1 Pa per 1 mV DC and a pressure range of +-2000 Pa between the differential pressure levels. It also features a maximum differential pressure of 75 kPa before the sensor breaks. These sensors feature an output level of less than 5 V DC. It has been connected directly to the controller without amplification or other signal modification. The pressure sensor accuracy has been defined as +- 6.25 % depending on environmental influences such as mechanical stress, temperature or humidity. The gain of 1000 mV per 1000 Pa differential pressure is not affected by environmental conditons.

One of each sensors inputs has been connected to one hole on the probe tip repectively. The other sensor input has not been connected and senses the atmospheric pressure (Figure 3-7 and Figure 3-8).

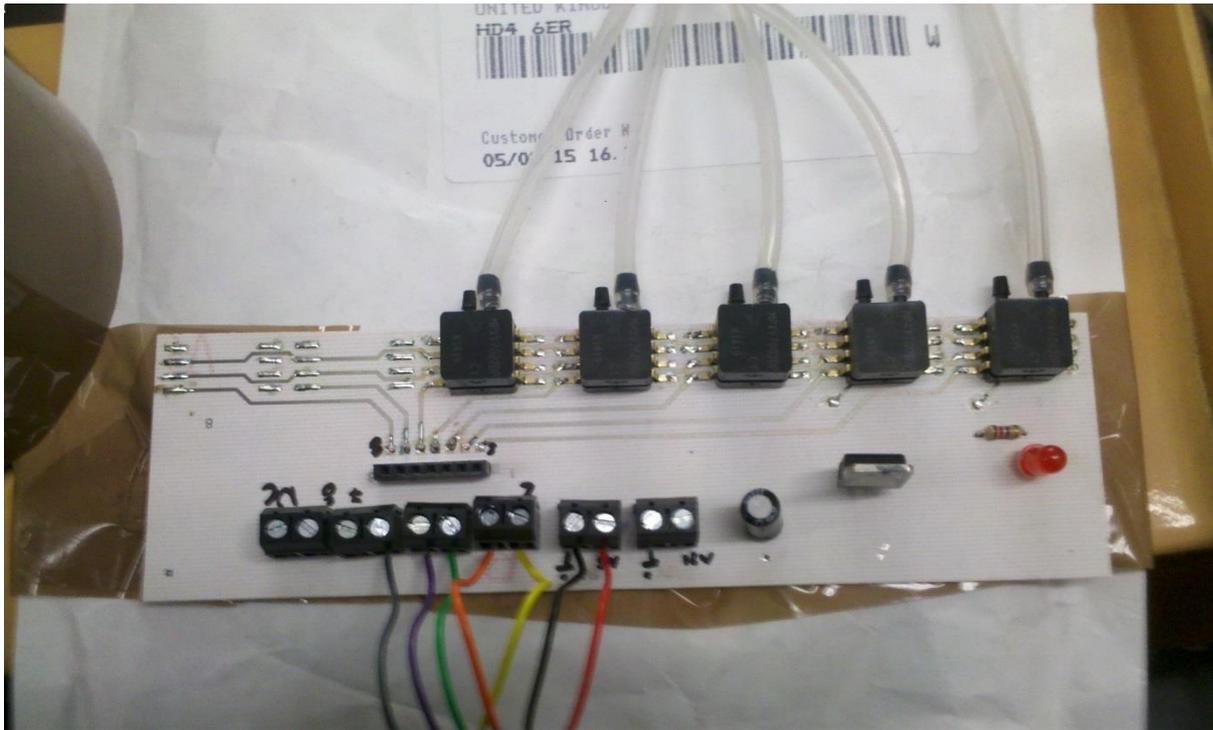


Figure 3-7: Pressure sensor board with five mounted sensors type MPXV7002



Figure 3-8: Five hole probe mounted on the traverse in the wind tunnel

A set of sensor output readings of all five pressure sensors allows to calculate the wind velocity vector that is facing the five hole probe tip. To calculate accurate wind information it is vital that the output signals of the pressure sensors are read simultaneously, which had to be considered in the process of the software development (chapter 5).

3.5 Five hole probe calibration

Each set of calibration parameters is unique for each probe. The methods for multi hole probe calibration can be split into two ideas: nulling methods and non-nulling methods. To utilize nulling methods the probe must be turned vertically (pitch), horizontally (yaw) and rotated inside the calibration wind tunnel, which requires a complex probe traversing mechanism. Non-nulling methods can be used with more simple traversing mechanisms. The probe only has to be turned vertically (pitch) and horizontally (yaw) in the wind tunnel. Therefore a non-nulling calibration method has been chosen.

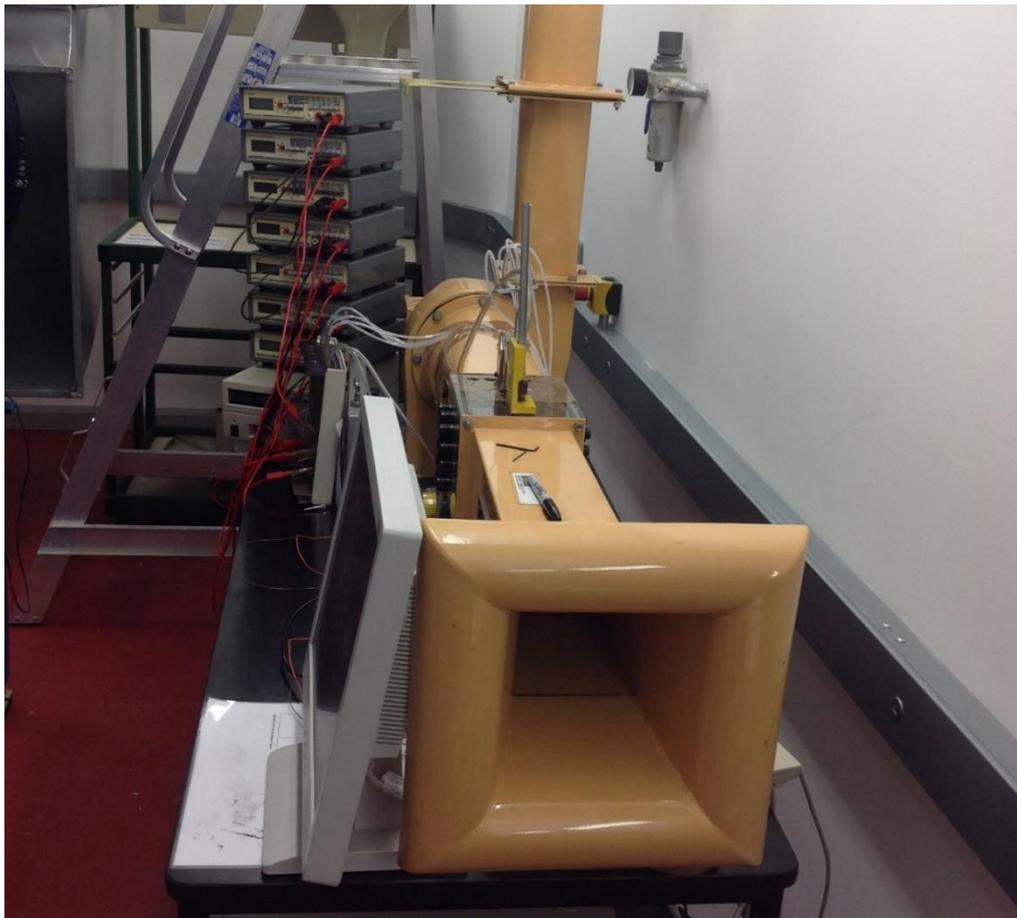


Figure 3-9: Wind tunnel setup for five hole probe calibration

Figure 3-9 shows a picture of the wind tunnel that has been used for wind condition simulation. The chosen wind tunnel has been a Plint & Partners TE93 with a (WxHxD) 120 mm x 120 mm x 1200 mm flow domain. The maximum flow velocity of the wind tunnel has been about 20 m/s or 45 mph. The centre element of the wind tunnels flow domain has been made of acrylic glass. Figure 3-10 shows the acrylic glass element that has been modified to include the pitch and yaw adjustable traversing mechanism for the five hole probe.

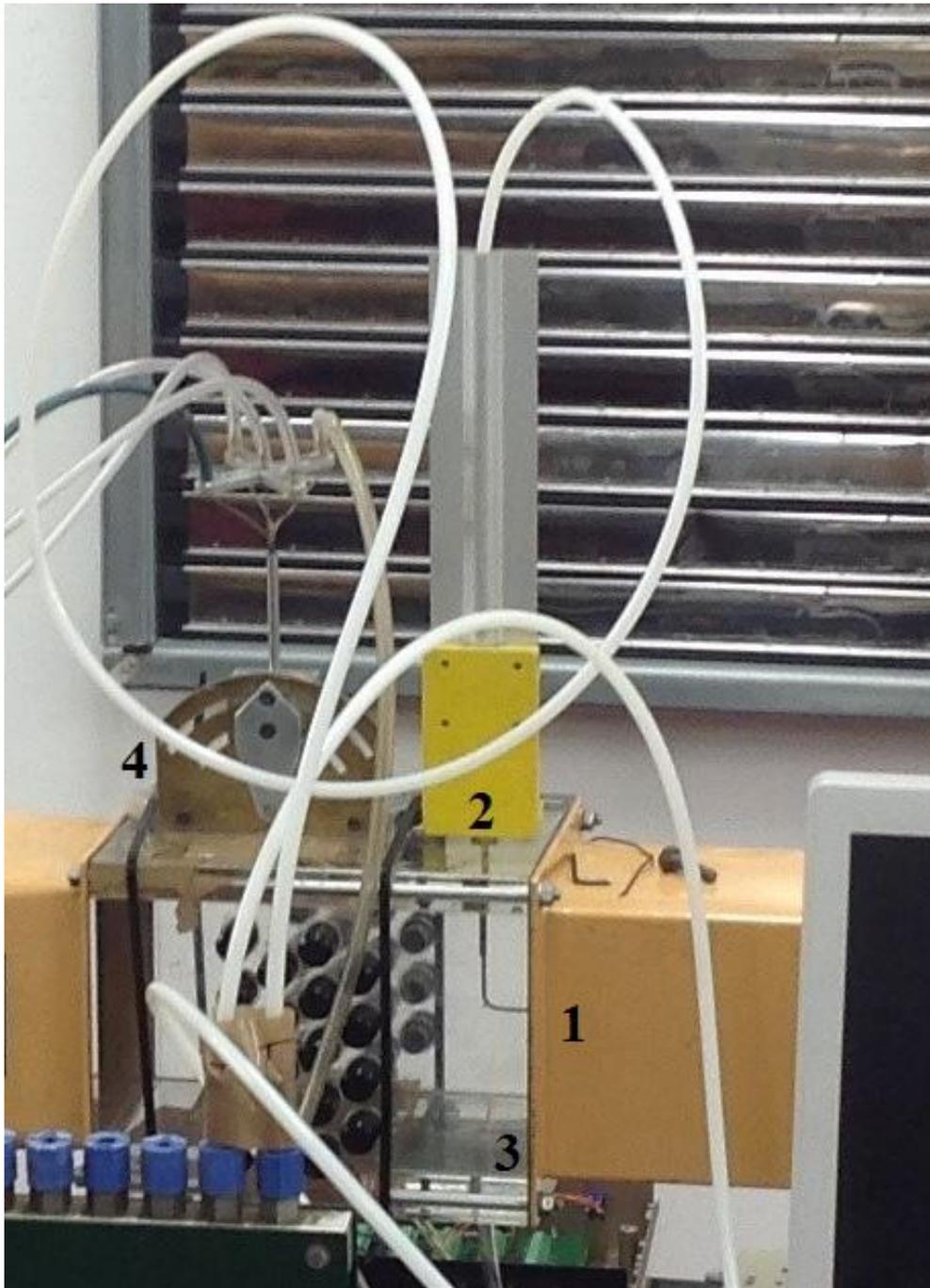


Figure 3-10: Picture of the wind tunnel centre flow domain element (not scaled)

Figures 3-10 and 3-11 show a picture and a schematic of the wind tunnels flow domain that has been used for the five hole probe calibration.

The following components can be seen:

1. Wind tunnel flow domain
2. Built in pitot tube tip for P_{total}
3. Built in pitot tube tip for P_{static}
4. Traverse (adjustable in pitch and yaw direction) with mounted five hole probe for non-nulling calibration method

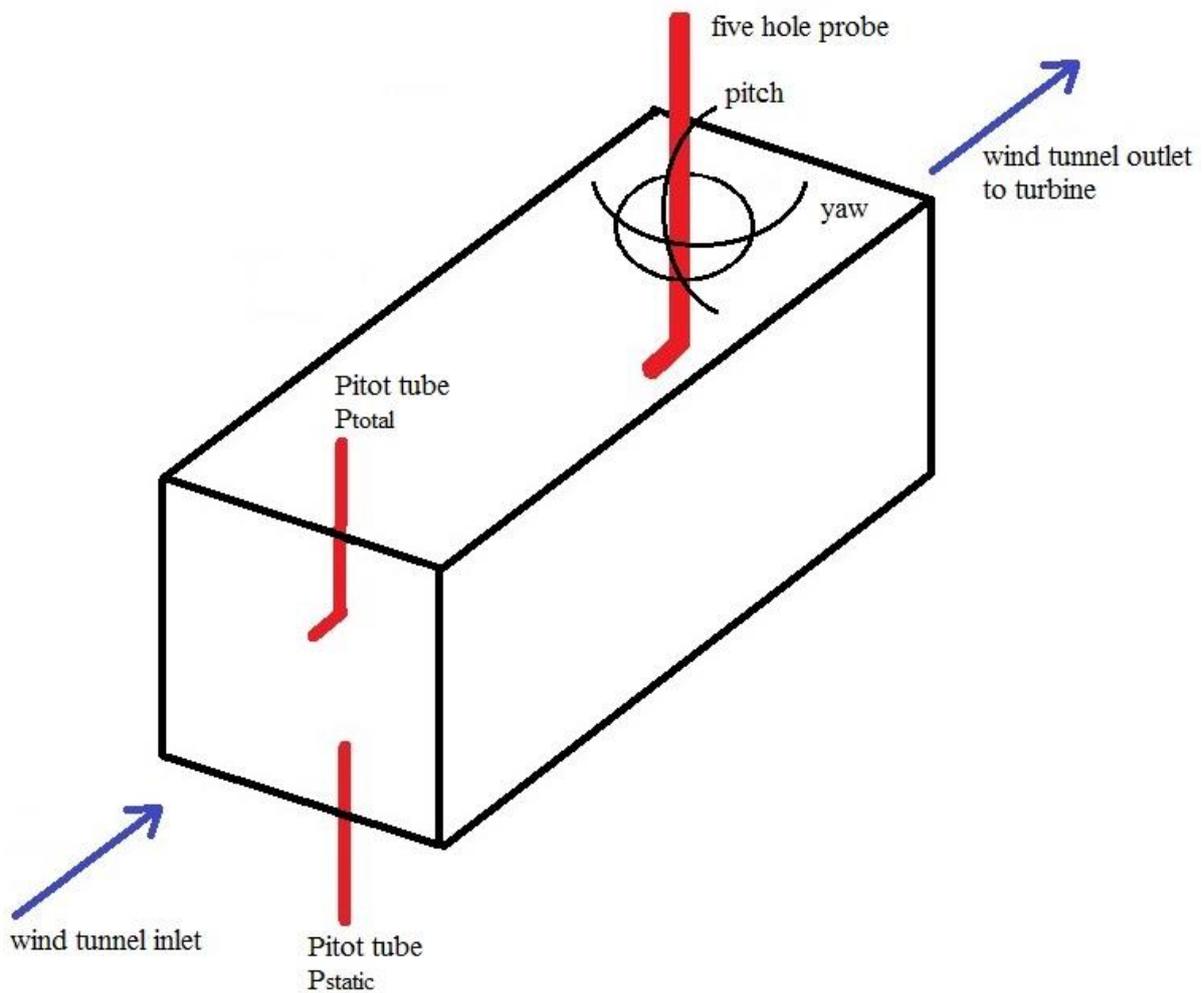


Figure 3-11: Schematic of the wind tunnel centre flow domain element (not scaled)

The probe has been mounted in the wind tunnel at position 50 mm from the bottom, so that it was in free stream just above the wind tunnels boundary layer. The mounting point of the probe

in zero pitch – zero yaw position had to be as low as possible because the probe tip is moving upwards during the calibration when the pitch angle is changed.

To apply non-nulling calibration methods a full set of calibration pressures has been acquired. One set of calibration pressures consists of two calibration angles of pitch and yaw, five pressure readings taken from each probe holes respectively and two pressure readings of P_{static} and P_{total} . The five pressure readings P_1 to P_5 have been acquired from the mounted five hole probe. The pressure readings of P_{static} and P_{total} have been acquired from the built-in pitot tube in the calibration wind tunnel. The calibration dataset is completed, when a set of seven pressure readings total has been taken for any combination of pitch and yaw angle within the range of calibration angles. The range of calibration angles and the chosen interval in angles may vary with different applications. A number of different mathematical data reduction techniques have been developed by different researchers for five hole probe research.

A. R. Paul et al. [26] proposed a refinement of non-nulling calibration method for a five hole probes, that proves to be the overall most accurate calibration process for five hole probes. The method uses the idea of sector wise data processing (see figure 3-12), which classifies the sets of pressure readings by the highest pressures among the probe pressures P_1 to P_5 into sectors 1 to 5 respectively. The further processing for the datasets is different for each sector, because the orientation of the probe in the flow field is considered. Therefore the calibration dataset is rearranged by sectors to simplify further processing.

Figure 3-12 shows the sector map than has been created after the calibration datasets have been marked and sorted by the highest pressures among P_1 to P_5 . For each combination of reference pitch angle and reference yaw angle the highest pressure has been noted in the map with the pressures number. If a calibration dataset has two equal highest pressures both pressures have been noted in the sector map, as it can be seen in the white fields. After all fields have been filled the map was coloured for clearness. It can be seen, that the distribution of the highest pressures per reading follows the scheme of the sensor nomenclature on the five hole probe tip (Figure 3-5). Compliance of the sector map and the probe tip nomenclature is a first sign for a successful calibration, though the precision cannot be determined by just a sector map.

	-42	-37	-32	-27	-22	-17	-12	-7	-2	3	8	13	18	23	28	33	38	43	48	yaw	
-48																					
-43	4	4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
-38	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2		
-33	4	4	4	14	1	1	1	1	1	1	1	1	1	1	1	2	2	2			
-28	4	4	4	4	4	1	1	1	1	1	1	1	1	1	2	2	2	2	2		
-23	4	4	4	4	4	14	1	1	1	5	5	5	1	12	2	2	2	2	2		
-18	4	4	4	14	1	1	1	15	15	5	1	1	1	2	2	2	2	2	2		
-13	4	4	4	4	4	4	5	5	5	5	5	5	2	2	2	2	2	2	2		
-8	4	4	4	4	4	4	5	5	5	5	5	5	25	2	2	2	2	2	2		
-3	4	4	4	4	4	4	5	5	5	5	5	5	2	2	2	2	2	2	2		
2	4	4	4	4	4	45	5	5	5	5	5	5	25	2	2	2	2	2	2		
7	4	4	4	4	4	5	5	5	5	5	5	25	2	2	2	2	2	2	2		
12	4	4	4	4	4	5	5	5	5	5	5	5	5	2	2	2	2	2	2		
17	4	4	4	4	4	45	5	5	5	5	5	5	25	2	2	2	2	2	2		
22	4	4	4	4	4	3	3	3	3	35	3	5	25	2	2	2	2	2	2		
27	4	4	4	4	3	3	3	3	3	3	3	35	25	2	2	2	2	2	2		
32	4	4	4	34	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2		
37	4	4	4	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2		
42																					
pitch																					

Figure 3-12: Sector Map created from the calibration data

The lowest pressure reading in a pressure set opposite each sector determining highest pressure is excluded from the calculation of the pressure normalization coefficient. That approach leads to a number of related but different calibration equations that slightly vary for each sector. It can be found, that the angle dependent coefficients follow the same idea of evaluating the opposing side pressures to determine angle information, as it is used for three hole probes (equation 5, equation 6).

The calibration equations (equation 6) have been used to calculate the calibration coefficients C_{P_α} , C_{P_β} , $C_{P_{total}}$ and $C_{P_{static}}$ for each dataset with respect to the sector each dataset belongs to. Following the suggestion of A.R. Paul a 4th order linear regression has been employed on the calibration data as data reduction method [26]. Each 4th order regression analysis has been applied separately for each sector, resulting in 5 times 4 sets of calibration

parameters. The calibration parameter sets are distributed on five sectors with four sets of calibration parameters for C_{P_α} , C_{P_β} , $C_{P_{total}}$ and $C_{P_{static}}$ respectively.

Zone-1:	Zone-2:	Zone-3:	Zone-4:	Zone-5:
$\bar{P} = \frac{P_4 + P_5 + P_2}{3}$	$\bar{P} = \frac{P_1 + P_3 + P_3}{3}$	$\bar{P} = \frac{P_4 + P_5 + P_2}{3}$	$\bar{P} = \frac{P_1 + P_3 + P_3}{3}$	$\bar{P} = \frac{P_1 + P_2 + P_3 + P_4}{4}$
$D = P_1 - \bar{P}$	$D = P_2 - \bar{P}$	$D = P_3 - \bar{P}$	$D = P_4 - \bar{P}$	$D = P_5 - \bar{P}$
$C_{P_\alpha} = \frac{P_1 - P_5}{D}$	$C_{P_\alpha} = \frac{P_1 - P_3}{D}$	$C_{P_\alpha} = \frac{P_5 - P_3}{D}$	$C_{P_\alpha} = \frac{P_1 - P_3}{D}$	$C_{P_\alpha} = \frac{P_1 - P_3}{D}$
$C_{P_\beta} = \frac{P_2 - P_4}{D}$	$C_{P_\beta} = \frac{P_2 - P_5}{D}$	$C_{P_\beta} = \frac{P_2 - P_4}{D}$	$C_{P_\beta} = \frac{P_5 - P_4}{D}$	$C_{P_\beta} = \frac{P_2 - P_4}{D}$
$C_{P_{total}} = \frac{P_1 - P_{total}}{D}$	$C_{P_{total}} = \frac{P_2 - P_{total}}{D}$	$C_{P_{total}} = \frac{P_3 - P_{total}}{D}$	$C_{P_{total}} = \frac{P_4 - P_{total}}{D}$	$C_{P_{total}} = \frac{P_5 - P_{total}}{D}$
$C_{P_{static}} = \frac{\bar{P} - P_{static}}{D}$				

Equation 6: Pressure normalisation coefficients for five hole probe by A.R. Paul et al. [26]

For the calibration of the five hole probe, the probe has been mounted on the traverse in the wind tunnel and adjusted to be facing the wind tunnel inlet (Figures 3-10, 3-11). The probe tube connectors have been connected to the pressure sensors type MPXV7002 (see Chapter 3.4 for details). The sensor output voltage levels have been measured with a number of multimeters type Black Star 3225 with 20 V DC input range selected.

To get accurate differential pressure from the sensors the sensor output voltages at zero differential pressure were measured and noted. These sensor output voltages at zero differential pressure have been used later on as reference voltages for balanced pressure condition. The pressure sensors P_1 to P_5 , P_{static} and P_{total} reference voltages at 0 kPa differential pressure have been measured. The range of these reference voltages was 2.25 V DC to 2.28 V DC.

The wind tunnel reference flow speed has been measured with a differential hand manometer Type Dwyer Series 475 Mark III, which has been connected to the wind tunnels built in Pitot tube. The measured differential pressure on the Pitot tube for the wind tunnel running on maximum velocity was 0.26 kPa, which is equal to a wind velocity of 20.6 m/s (Equation 3). To find the reference zero- zero position for pitch and yaw angle of the five hole probe the pressures on holes 1-3 and 2-4 has been balanced. Therefor the five hole probe has been mounted on the traversing mechanism and inserted into the wind tunnel. The five hole probe tube connectors have been connected to one pressure sensor P_1 to P_5 each. The power supply for the pressure sensors and the wind tunnel turbine have been switched on. The five hole probe has been rotated in yaw direction until the pressures 2-4 were balanced (the difference in the

pressure sensor outputs for 0 kpa differential pressure have been respected). The yaw reference angle has been noted from the traversing mechanism. To find the pitch reference position the probe has been tilted in pitch direction until the pressures 1-3 were balanced. The pitch reference angle has been noted from the traversing mechanism. The pressures on 1-3 were balanced at +3 ° pitch and pressures 2-4 were balanced at -3 ° yaw respectively. These mechanical maladjustments have been taken into account during the calibration compiling later on. The implementation of the zero-zero position offsets of +3° and -3° also can be seen in the sector map in Figure 3-12.

One calibration dataset contains a combination of two angles for pitch and yaw, five pressures for P_1 to P_5 and two pressures P_{total} and P_{static} obtained from the built in Pitot tube. The probe has been calibrated from -40 ° to +40 ° each for pitch and yaw combination in 5 ° increments each, with seven pressures/ voltages for P_1 to P_5 and P_{total} and P_{static} respectively. These datas have been transferred and processed in Microsoft Excel. The full dataset including different stages of the calibration process can be found on DVD in appendix A5.

3.5.1 Step wise description for the calibration of a five hole probe

1. Full Calibration data with sensor output voltages as described in chapter 3.1 has been transferred into a Microsoft Excel worksheet
2. Pitch and yaw angles of each dataset have been edited to compensate maladjustments from mounting the probe for calibration (+3 ° pitch and -3 ° yaw)
3. The sensor reference values for 0 kPa differential pressures have been subtracted from the sensor output values.
4. Voltage to pressure conversion has been applied to all datasets, assuming atmospheric pressure (zero differential pressure) as 101300 Pa and the given gain for the pressure sensors of 1000 Pa per 1000 mV.
5. The highest absolute pressure for each dataset in the calibration data has been highlighted and the calibration data sorted by the highlighted highest pressure per dataset (sector wise sorting)
6. Calculation of the calibration coefficients C_{P_α} , C_{P_β} , $C_{P_{total}}$ and $C_{P_{static}}$ has been performed for the entire calibration data. The equations used to calculate these

coefficients are dependent on the related sector. The equations can be found in equation 6.

7. Multiples of the calibration coefficients C_{P_α} and C_{P_β} have been calculated as preparation for a 4th order linear regression analysis.
8. The linear regression analysis has been performed using the built in LINEST function of Microsoft Excel. This calibration step has been performed for each combination of the calibration parameters pitch angle α , yaw angle β , pressure coefficient $C_{P_{static}}$ and pressure coefficient $C_{P_{total}}$ and each sector, resulting into 4 times 5 sets of calibration parameters (4 calibration coefficients C_{P_α} , C_{P_β} , $C_{P_{total}}$ and $C_{P_{static}}$ times 5 sectors).

Excels LINEST [27] function outputs an array with a number of parameters that best fit a polynomial equation to describe the input data as an equation, following the least square method. The goodness of fit value of each calibration parameter is marked as “r²” value, representing the deviations of the scattered parameters from the best fitted curve. An r² value of 1 represents a perfect fit where all scattered parameters exactly fit on the calculated curve and r² values significantly less than 1 represent a bad curve fitting big deviations between the best fit curve and the scattered parameters. The obtained 5 times 4 calibrations sets of calibration parameters have an accuracy range of 70 % to 98 % and one deviating set with 24 %. The distribution of precision per calibration set is listed in the table 3-1. The full tables with the calibration parameters can be found in the appendix A1 or electronically on DVD in appendix A5 (The LINEST function is named RGP in German MS Office 2013 package [28]. The excel sheet provided on DVD in Appendix A5 may show LINEST or RGP).

r ²	Sector 1	Sector 2	Sector 3	Sector 4	Sector 5
Pitch angle α	0.88	0.95	0.94	0.97	0.84
Yaw angle β	0.71	0.93	0.97	0.91	0.98
$C_{P_{static}}$	0.97	0.96	0.96	0.93	0.24
$C_{P_{total}}$	0.97	0.92	0.74	0.79	0.70

Table 3-1: Overview goodness of fit for calibration parameters

The obtained calibration parameters have been used to calculate wind condition information from pressure readings obtained from the calibrated five hole probe.

3.6 Calculation of wind information from a single set of five hole probe pressure readings

To measure the wind information from a set of obtained pressure readings from the calibrated five hole probe the pressure readings have been combined with the matching calibration parameters. But the pressure readings cannot be combined with the calibration parameters directly. The obtained pressure readings of P_1 to P_5 have been checked for the highest pressure value among them to determine the sector of the set of pressure readings.

With known sector (Figure 3-12) the calibration parameters C_{P_α} and C_{P_β} have been calculated, following the same equations that have been used during the calibration process (Equation 6). The calibration parameters calculated from the pressure sensor outputs and the calibration parameters $a_0 - a_{14}$ have been combined in a generic 4th order equation 7 (the nomenclature of that equation was taken from the work of A. R. Paul [26]. The nomenclature in the help file of Microsoft Excel [27] which explains the structure of the output array of the LINEST function diverts from that nomenclature as follows: $a_0 = b$, $a_1 = m_1$, ..., $a_n = m_n$):

$$f(c_{p_\alpha}, c_{p_\beta}) = a_0 + a_1 c_{p_\alpha} + a_2 c_{p_\beta} + a_3 c_{p_\alpha}^2 + a_4 c_{p_\beta}^2 + a_5 c_{p_\alpha} c_{p_\beta} + a_6 c_{p_\alpha}^2 c_{p_\beta} + a_7 c_{p_\alpha} c_{p_\beta}^2 + a_8 c_{p_\alpha}^3 + a_9 c_{p_\beta}^3 + a_{10} c_{p_\alpha}^2 c_{p_\beta}^2 + a_{11} c_{p_\alpha}^3 c_{p_\beta} + a_{12} c_{p_\alpha} c_{p_\beta}^3 + a_{13} c_{p_\alpha}^4 + a_{14} c_{p_\beta}^4$$

Equation 7: 4th order equation for the calibration of a 5 hole probe

The output of equation 7 changes depending on what set of calibration parameters have been inserted in addition to the C_{P_α} and C_{P_β} values. Inserting the set of calibration parameters for pitch angle α outputs the actual pitch angle α , inserting the calibration parameters for yaw angle β outputs the actual yaw angle β . Inserting the calibration sets of the pressure coefficients $C_{P_{total}}$ and $C_{P_{static}}$ results in those pressure coefficients, which have to be further processed to calculate the total pressure P_{total} and the static pressure P_{static} . To do so the equations for $C_{P_{static}}$ and $C_{P_{total}}$ from equation 6 have been solved for P_{static} and P_{total} for each sector respectively. The resulting equations can be seen in equation 8.

Zone-1 :

$$\bar{P} = \frac{P_2 + P_4 + P_5}{3}$$

$$P_{total} = -C_{P_{total}} * (P_1 - \bar{P}) + P_1$$

$$P_{static} = -C_{P_{static}} * (P_1 - \bar{P}) + \bar{P}$$

Zone-2:

$$\bar{P} = \frac{P_1 + P_3 + P_5}{3}$$

$$P_{total} = -C_{P_{total}} * (P_2 - \bar{P}) + P_2$$

$$P_{static} = -C_{P_{static}} * (P_2 - \bar{P}) + \bar{P}$$

Zone-3:

$$\bar{P} = \frac{P_2 + P_4 + P_5}{3}$$

$$P_{total} = -C_{P_{total}} * (P_3 - \bar{P}) + P_3$$

$$P_{static} = -C_{P_{static}} * (P_3 - \bar{P}) + \bar{P}$$

Zone-4:

$$\bar{P} = \frac{P_1 + P_3 + P_5}{3}$$

$$P_{total} = -C_{P_{total}} * (P_4 - \bar{P}) + P_4$$

$$P_{static} = -C_{P_{static}} * (P_4 - \bar{P}) + \bar{P}$$

Zone-5:

$$\bar{P} = \frac{P_1 + P_2 + P_3 + P_4}{4}$$

$$P_{total} = -C_{P_{total}} * (P_5 - \bar{P}) + P_5$$

$$P_{static} = -C_{P_{static}} * (P_5 - \bar{P}) + \bar{P}$$

Equation 8: Pressure calculation equations for P_{total} and P_{static} with respect to sectors

The resulting pressures P_{total} and P_{static} can be used to calculate the wind velocity with equation 3.

3.6.1 Step wise description for the calculation of wind information from a set of five hole probe pressure readings

1. A set of five pressure readings from a five hole probe consisting of P_1 to P_5 has been taken.
2. The sector of the set of pressure readings has been determined by finding the highest pressure among the readings.
3. Equation 6 has been used to calculate the pressure coefficients C_{P_α} and C_{P_β} with respect to the sector.
4. Pitch angle α and yaw angle β have been calculated by combining the related sets of calibration parameters a_0 to a_{14} with respect to the sector and the pressure coefficients C_{P_α} and C_{P_β} in equation 7.
5. Pressure coefficients $C_{P_{total}}$ and $C_{P_{static}}$ have been calculated by combining the sets of calibration parameters a_0 to a_{14} with respect to the sector and the pressure coefficients C_{P_α} and C_{P_β} in equation 7.

6. The pressures P_{total} and P_{static} have been calculated from the pressure coefficients $C_{P_{total}}$ and $C_{P_{static}}$ from equation 8 with respect to the sector.
7. The wind velocity v has been calculated using equation 3.

The calibration process has been explained and presented, also calculation of wind condition information based on five hole probe readings and calibration parameters has been presented. The following section addresses the validation and the reliability of the calibration and calculation results.

3.7 Validation of the calibration results

The validation of calibration and calculations follows a basic concept: If the calibration and the algorithm are working correctly, angles and velocity of the wind vector can be calculated from the corresponding pressures P_1 to P_5 for each dataset. Therefore the full set of raw data that was taken during the beginning of the calibration process has been used as oracle for the calibration process. The calculation process as described in chapter 3.5 and 3.5.1 has been applied to each dataset in the calibration data, using the calibration datasets of P_1 to P_5 as inputs and the corresponding calibration wind conditions as reference outputs (see Figure 3-13). Full Validation data is provided electronically. It can be found in Appendix A4.

reference angles		algorithm inputs								
pitch alpha	yaw beta	Top p1	Right p2	Bottom p3	Left p4	Center p5	Total pt	Static ps		
-3	3	101200	101180	101170	101170	101300	101300	101050		
-3	8	101190	101220	101180	101120	101290	101300	101050		
-3	13	101160	101240	101170	101080	101290	101300	101050		
-3	18	101160	101270	101140	101050	101260	101300	101050		
-3	23	101130	101290	101100	101010	101240	101300	101050		
-3	28	101110	101280	101060	100970	101220	101300	101050		
-3	33	101070	101290	101010	100920	101150	101300	101050		
-3	38	101020	101290	100960	100890	101110	101300	101050		
-3	43	100960	101290	100920	100860	101040	101300	101050		
-3	48	100930	101280	100860	100830	100980	101300	101050		
-3	-2	101190	101150	101180	101200	101300	101300	101050		
-3	-7	101170	101120	101160	101220	101300	101300	101050		
-3	-12	101170	101070	101160	101260	101300	101300	101050		
-3	-17	101130	101030	101130	101280	101260	101300	101050		
-3	-22	101100	100990	101100	101280	101240	101300	101050		
-3	-27	101070	100940	101060	101280	101220	101300	101050		
-3	-32	101000	100890	101020	101300	101190	101300	101050		
-3	-37	100970	100840	100970	101300	101120	101300	101050		
-3	-42	100890	100800	100940	101290	101070	101300	101050		
2	3	101140	101170	101200	101150	101300	101300	101050		
2	8	101170	101240	101190	101130	101290	101300	101050		

Figure 3-13: Section of calibration raw data.

The calculation results and the oracle references have been visualized in the following Figures 3-14 to 3-16, giving information about how precise the wind information can be calculated from pressure values given by the calibrated five hole probe.

Figure 3-14 shows the reference pitch angles taken from the calibration dataset displayed as blue dots and the calculated pitch angles from the wind condition calculation output. The data has been sorted by reference pitch angle, starting with the lowest. Each reference value is plotted against the calculated value for each dataset, over a total of about 310 datasets. Based on the calculation results a trend line has been added to determine the precision of the calculated values. This also provides a graphical impression of the goodness of the calculation results. The r^2 value which shows the goodness of fit for the linear trend line with the scatters of calculation results. The closer the values comes to 1.00, the better is the match of the trend line with the calibration results. 1.00 represents a calibration precision of 100 %. The r^2 for the calibration in yaw direction is 0.97, which leaves an estimated error of about 3 %.

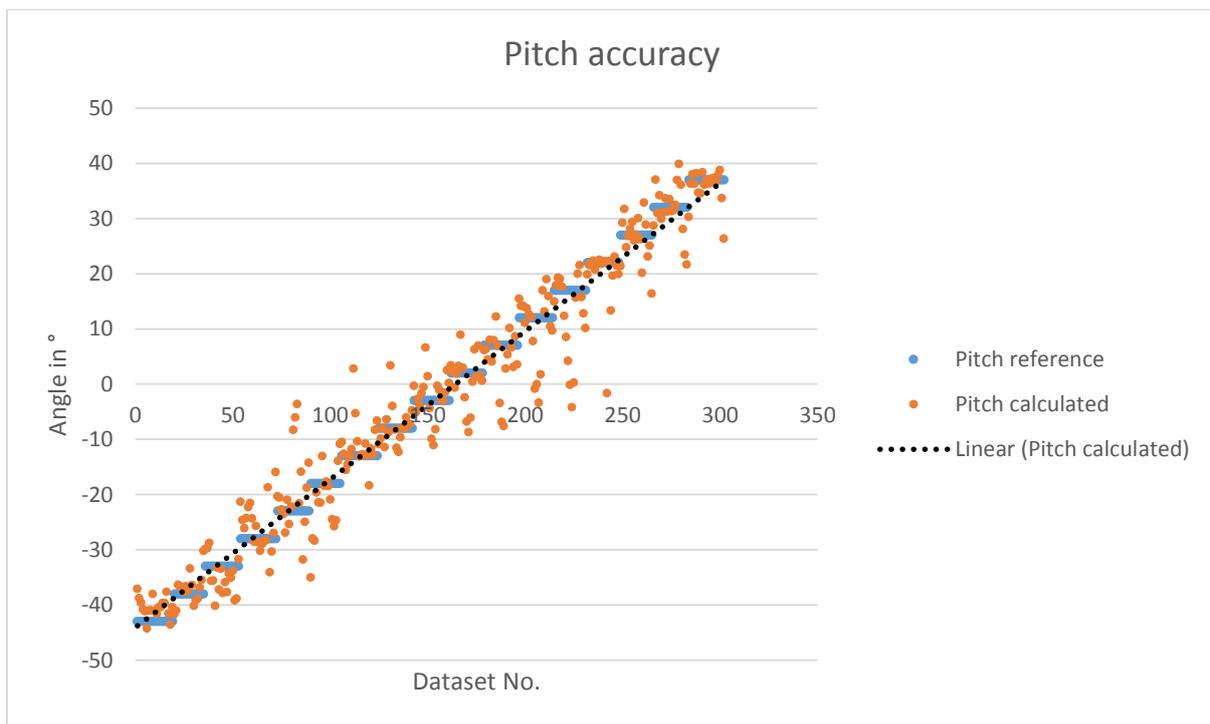


Figure 3-14: Accuracy for five hole probe calibration in pitch direction

Figure 3-15 shows a plot of reference yaw angles against calculated yaw angles. The data has been sorted by reference yaw angle, starting with the lowest. A linear trend line has been added

to determine the precision of the calculated values. The r^2 value for the linear trend line is 0.98, which leaves an estimated error of about 2 %.

For both figure 3-14 and figure 3-15 it can be seen, that the linear regression lines for calculated values are running through the midsection of each blue section of the reference angle graph. This supports the good result of calibration process and the wind information calculation.

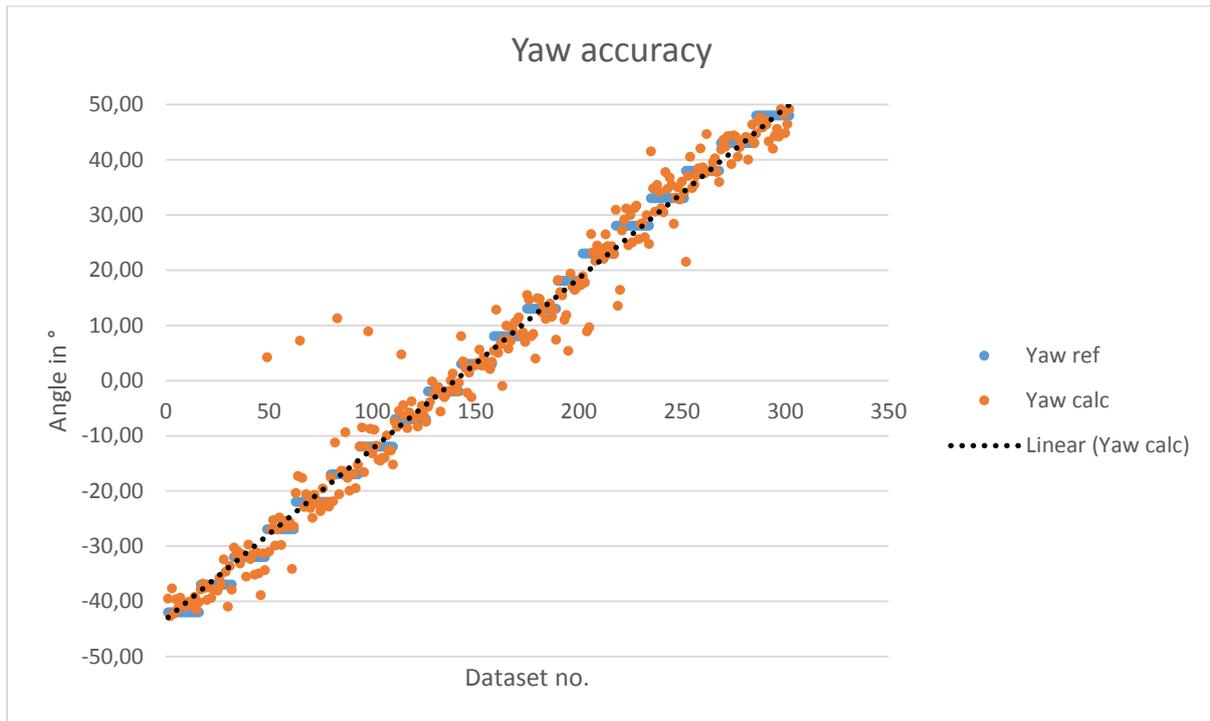


Figure 3-15: Accuracy for five hole probe calibration in yaw direction

Figure 3-16 shows the calibration results of the wind velocity measurement. The reference wind speed has been calculated (equation 2) from the measured differential pressure that has been taken from the wind tunnels built in Pitot tube and is 20.2 m/s (Figure 3-14). The wind velocity calculation results fluctuate around the value of 20.2 m/s but also does the dataset contain some scattered values that are too far of the majority of the values to be ignored.

Tracing the scattered values in the wind velocity table through the calibration process does not give a reasonable explanation why the scatters exist. Neither a flaw in the calibration parameters, a flaw in the calibration process, an error in the calibration raw data or an error in the calculations has been found. This leads to the assumption, that the calibration data which has been used as oracle for the validation might contain few combination of faulty angle-pressure datasets. These faults are likely related to misreadings within the pressure values of

P_1 to P_5 during the calibration process. However Figures 3-14 to 3-16 show that the precision of the overall calibration process is not affected.

The averaged calculated wind velocity is at 20.3 m/s and only differs by 1 % from the reference reading of 20.2 m/s. This strongly supports the thesis of fault in single pressure readings within the calibration data. Consequently these faulty pressures result in mismatched pressure-angle relations in the calibration data that has been used as reference dataset. The overall effect of these faults has been minimised by the linear regression that was used for data reduction in the calibration process.

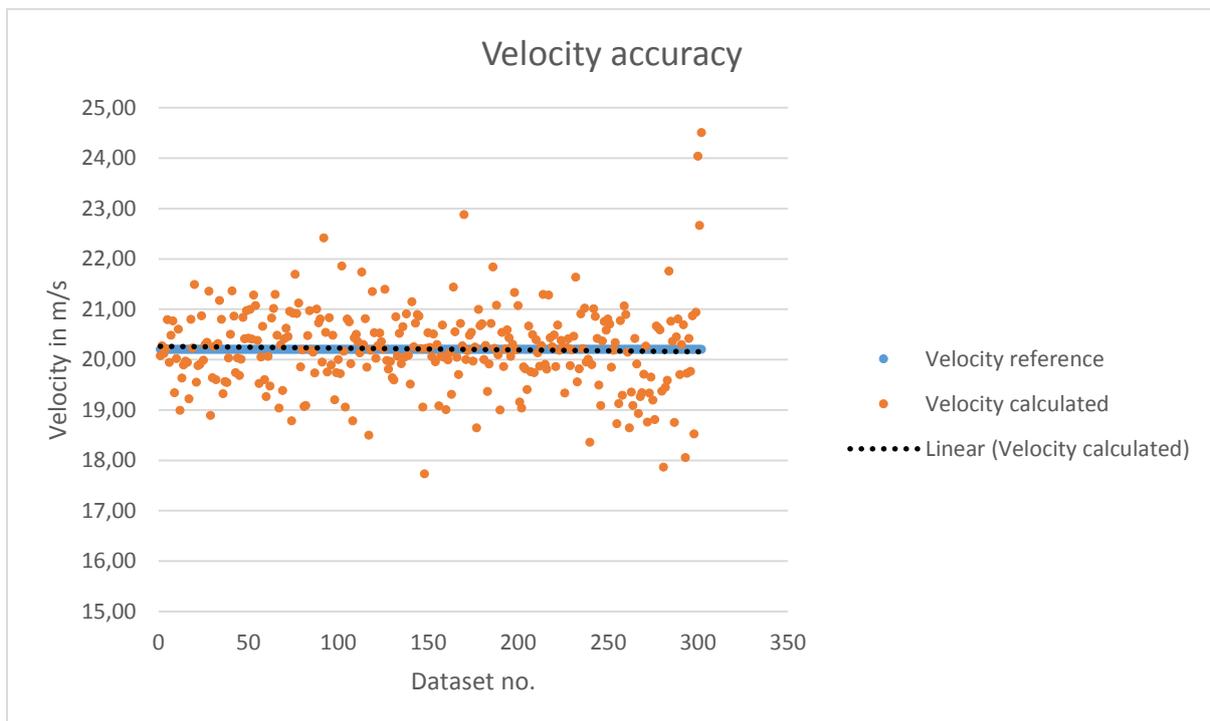


Figure 3-16: Accuracy for five hole probe calibration in wind magnitude

It seems that the overall calibration accuracy has not been affected by single faults in the calibration raw data. The obtained precisions for pitch angle calculation are about 97 % (Figure 3-14), precision for yaw angle are about 98 % (Figure 3-15) and the precision for the wind velocity calculation is about 99 % (Figure 3-16). The datasets and graphs of Figures 3-14 to 3-16 can be found on DVD in Appendix A5.

3.8 Conclusion

The adaption of known technology in the area of pressure based wind condition measuring has been presented. This includes the different pressure probes for:

- velocity measurement with a pitot tube

- two dimensional measurement of velocity and one planar angle with a three hole probe
- three dimensional measurement of velocity and two orthogonal planar angles with a five hole probe

A method for calibration of five hole probes has been presented and performed. Also the method to calculate wind information based on five hole probe pressure sensor readings and calibration parameters has been presented. The evaluation of calibration and wind condition calculation shows, that even with some errors in the calibration data good calibration results with more than 97 % overall accuracy have been achieved.

The pressure sensors for the given application have been chosen by the estimated wind and pressure condition. Differential pressure sensors type Freescale Semiconductor MPXV7002 with an output signal of 0.5 V DC to 4.5 V DC and a gain of 1000 mV DC per 1000 Pa pressure difference on the inputs have been chosen.

With technology adapted to sense wind information with a decent accuracy the first part of the defined research objectives has been accomplished. The following chapter describes the development of a sensor configuration, which is capable of detecting the deflector position and the trailer height. The outputs of the wind sensor and the CRD positioning sensors are merged and evaluated in hardware and software of a fitted control system.

Chapter 4

Development of a control circuit and a sensor array for the automation of an automated cab roof deflector

The previous chapter presented the characteristics of different pressure based flow probes for measuring wind vector information. A calibration method for multi hole probes and implementation of the calibration results in a control system for multi hole probes has been presented. Furthermore a method for validation of flow probe calibration results has been presented.

This chapter describes the dimensions of the cab roof deflector used in this project and the setup of the sensor array mounted on the deflector. It also describes the implementation of the actuator that was used for this project and the control system developed to create an operational system.

4. Development of a control circuit and a sensor array for the automation of an automated cab roof deflector

The cab roof deflector which has been used in the laboratory setup consists of two side deflector panels and a deflector shield. All components have been mounted on a table which represented the cab roof. The dimensions of the deflector are (HxWxD) 450 mm x 2000 mm x 870 mm with a deflector shield length of 920 mm (Figure 4-1). A linear actuator type Industrial Devices LTD IDM8A [29] with a self-locking 1:20 gear ratio has been mounted between table and the deflector shield. The chosen actuator operates on 24 V DC and therefore can be supplied directly by the trucks board net.

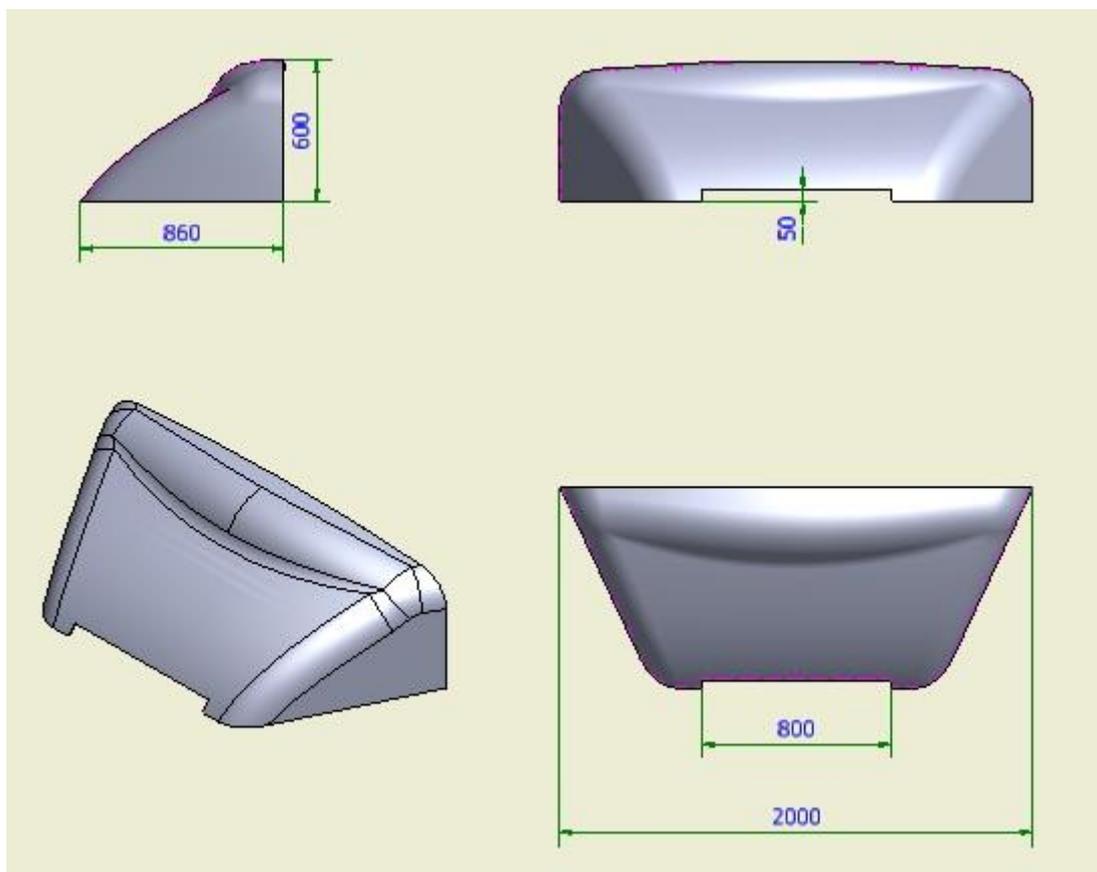


Figure 4-1: 3D model of the used deflector

To operate the deflector autonomously the system needs to acquire three types of information: The position of the deflector, the trailer reference height and the wind condition information. Therefore the system requires three different types of sensors. A five hole probe sensor has been used for reading the wind condition (see chapter 3.5). Two infrared distance sensors have been used for detection of the trailer height and measuring the deflector position respectively.

Figure 4-2 and Figure 4-3 show a schematic and a picture of the setup for the automated cab roof deflector system respectively. The absolute position for the deflector shield is measured by a distance sensor that has been mounted in vertical orientation on the cab roof. This sensor measures the distance between the cab roof and the deflector shield.

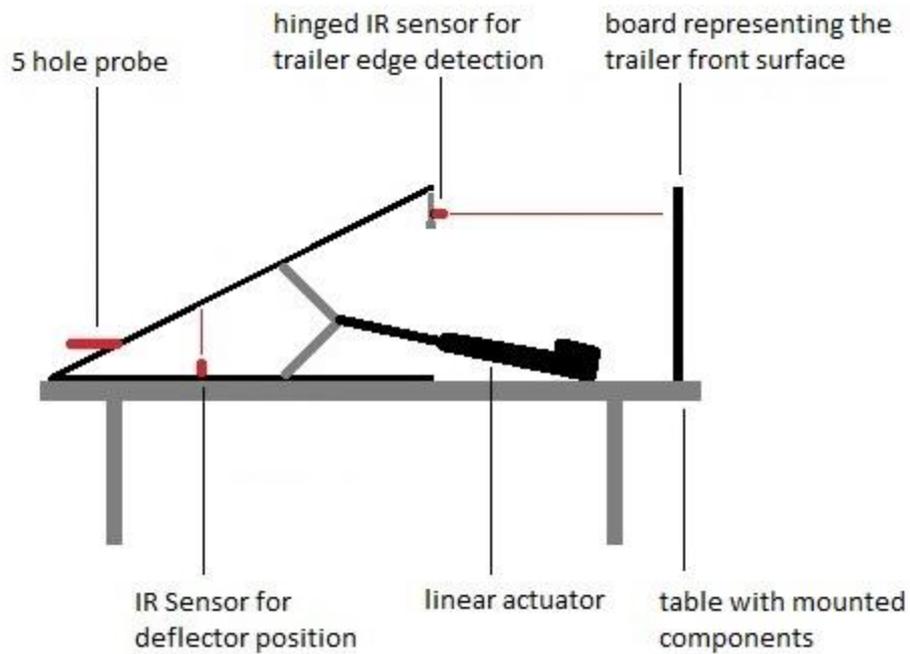


Figure 4-2: Schematic of the setup for the automated cab roof deflector system

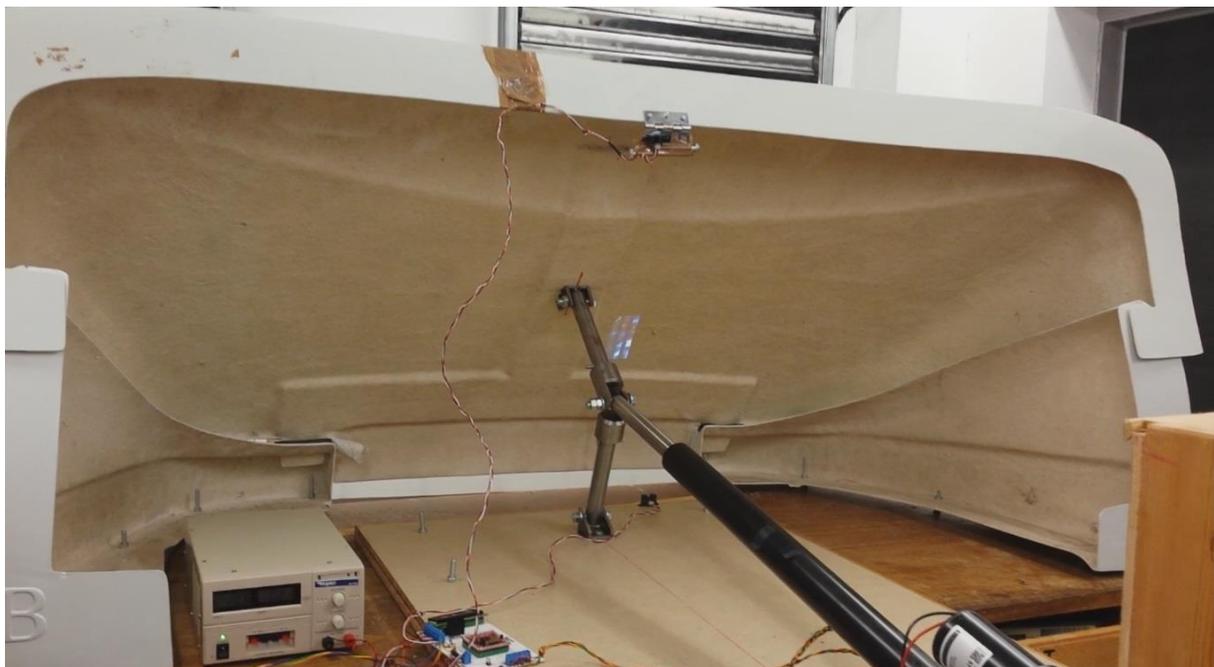


Figure 4-3: Picture of the setup for the automated cab roof deflector system

Another distance sensor has been mounted 115 mm below the top edge of the deflector. The sensors mounting bracket has been hinged and additionally weighted for keeping the sensors orientation horizontal to achieve a precise edge detection signal. The trailer front surface has been represented by a wooden board. The wooden board has been adjusted in height to simulate the varying trailer heights from 4.2 m to 4.8 m.

4.1 Selection of distance sensors

The previous section described the mechanical setup of the automated cab roof deflector system as well as positioning of the mounted sensors and actuator. The trailer height has been measured by a combined evaluation of the deflector position sensor and the trailer edge detection sensor. The deflector moves upwards so that the mounted edge detection sensor moves upwards too. When the edge detection sensor is moved above the trailer the sensor output signals “out of range”. Then the deflector is moved downwards until the edge detection signal outputs no more “out of range” and the deflector stops moving. The measured trailer position minus the 115 mm between sensor and trailer top edge are stored as trailer reference height.

The distance sensor for measuring the deflector position (Figure 4-2, 4-3) had to fulfil the following requirements to be suitable for the application. The requirements are based on the geometry and the make of the deflector used for the prototype system as well as the setup of the sensor array:

- The sensor needs to precisely measure distances in a range of about 20 cm to 50 cm for the measurement of the deflector position. This is required because the sensor is mounted under the deflector shield and measures the distance between the cab roof (represented by a table in this prototype) and the deflector shield. The distance between the sensor and the deflector shield is in the range of 20 to 50 cm for this prototype.
- The sensor is required to detect and measure distances to surfaces which are not orthogonal to the sensors direction of measurement but diverts in a range of 90 ° to 45 °. This is required because the deflector has an elliptical shape and is changing its position and the angle relative to the fixed sensor during automatic operation.

The distance sensor for detecting the trailer height had to fulfil the following requirements to be suitable for the application:

- The sensor needs to be suitable as a proximity switch with a detection distance of about 90 cm for the detection of the trailer (see dimensions of articulated trucks in

chapter 1). This is required because of the planned function of the edge detection: The deflector moves up and lifts the sensor over the trailer top edge. When the sensor outputs “out of range” the deflector moves back down. When the sensor detects the trailer top edge the sensor output will jump back to a value related to a distance. The sensor therefore needs to be able to measure the distance between the deflector and the trailer which is standardized to about 90 cm [3].

- The detection of the trailer edge requires that the distance sensor has a narrow angle of detection to enable precise edge detection

It was preferred to use the same sensor type for both applications of distance measuring and trailer detection if possible. For sensors of the same type the characteristic voltage output to distance conversion curve can be reused in the controller software.

All required sensor characteristics have been found with optical distance sensors. Therefore the same sensor type could be used for both applications. The chosen distance sensors for the system developed in laboratory environment (with static environmental conditions in terms of temperature and humidity) were two infrared distance sensors type SHARP GP2Y0A02YK0F [30].

Because both the deflector shape and the sensor output curve are non-linear it has been chosen to find an equation that describes the relationship between the sensor voltage output and the deflector position. Figure 4-4 shows the blue curve that represents the measured curve and the orange curve that represents the linearized sector curve which was implemented in the controller software for measuring the deflector height. The presented graph represents the deflector’s position as distance between the highest point of the deflector and the table where the system is mounted on, which represents the cab roof level of an articulated truck. To match the deflector positioning in absolute height above road level an offset has been added to represent the cab height seen from road level.

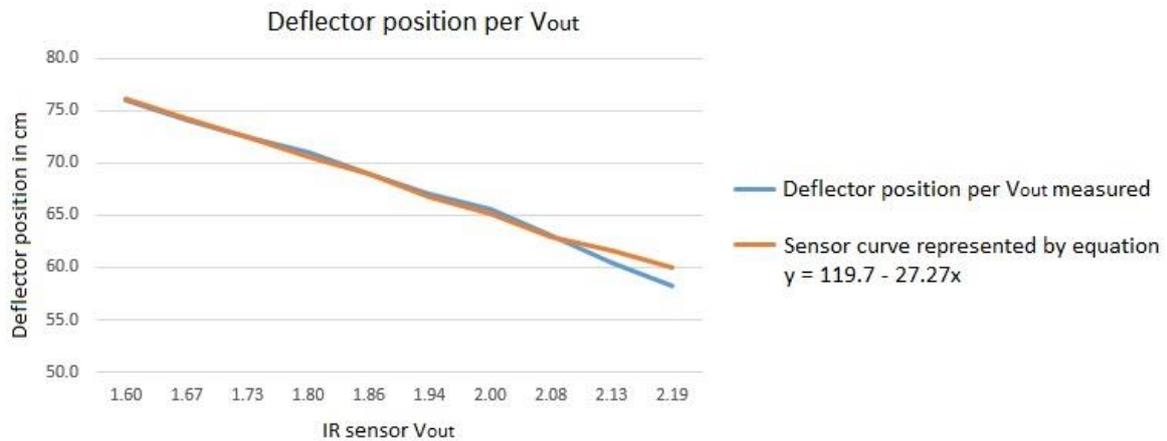


Figure 4-4: Deflector height per v_{out} of the used Sensor

The requirements in controller outputs and in system setup has been defined recently. With known requirements the microcontroller has been selected and the controller board has been designed.

4.2 Control system setup

With the deflector actuator and sensor specifications defined (Chapter 3.4 and Chapter 4.1) the microcontroller has been selected and the control system PCB has been designed. Tests have shown that the required starting current for the selected linear actuator is about 9 A. Because 24 V DC and 9 A cannot be controlled by a microcontroller directly it was necessary to implement a motor driver that can safely operate this much power. Therefore an H-Bridge motor controller type ST Microelectronics VN3SP30-E [31] has been implemented to drive the actuator. It can provide up to 30 A at up to 36 V DC and is controlled with 5 V inputs which are compatible to most microcontroller chips. The power supply for the chip is 24V DC and GND. The inputs are: EN enables the H-Bridge to operate, PWM can control the motor speed by reducing the effective DC voltage (steady 5V DC/high signal for steady 24V on the motor outputs), INA and INB for supplying 24V DC or GND on the motor outputs depending on the applied input signals. The motor is directly connected to the outputs OUTA and OUTB. As stated before the chip can supply 24V and GND on both outputs. Table 4-1 shows the resulting modes of operation:

Input INA	Input INB	Output OUTA	Output OUTB	Function
Low	Low	0 V DC	0 V DC	Passive brake
Low	High	0 V DC	24 V DC	Motor turn left
High	Low	24 V DC	0 V DC	Motor turn right
High	High	24 V DC	24 V DC	Active brake

Table 4-1: H-Bridge modes of operation

Figure 4-5 shows the PCB with mounted H-Bridge and the wire connectors.

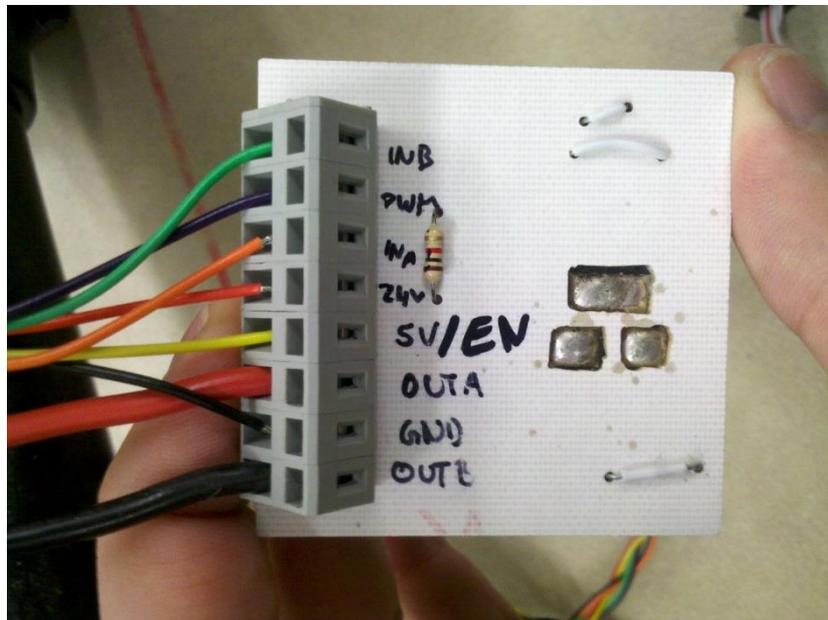


Figure 4-5: PCB with H-Bridge motor driver VNH3SP30-E

The controller had to feature at least the following specifications:

- 7 Analogue to digital converter (ADC) inputs for five pressure sensors and two IR distance sensors
- Adjustable ADC reference voltage with $V_{ref, max} = 5 \text{ V DC}$
- 4 binary outputs for controlling the h-bridge which drives the actuator
- 8 binary outputs for interfacing of a LCD Display for debugging purposes
- Supply voltage of 5 V DC

- Atmel Microcontroller preferred because of previous knowledge
- Preferably already used in automotive applications

The controller which fulfils the mentioned specification has been found with an Atmel AT90CAN128 8-Bit Microcontroller [32]. It features eight 10 bit ADCs for the pressure sensor inputs and the distance sensor inputs, configurable binary input-output pins for controlling the h-bridge and the debugging display as well as several communication interfaces. All input and output pins support a maximum voltage of 5 V DC. To increase the sensitivity on the ADC inputs the analogue reference voltage was reduced from standard 5 V DC to 3 V DC. The conversion of 24 V DC truck battery level to 5 V DC for the sensors and the controller supply is achieved with a Tracopower TSR1-2450 [33] switching voltage regulator.

4.3 Conclusion

The mechanical model that has been used for the research project on a control system for automated cab roof deflectors has been presented. This includes the dimensions of the used deflector, the used linear actuator and the used sensor positions for the wind information sensing. It also includes the positioning of the distance sensors for measuring the deflector position and the trailer height, as well as the conversion of their sensor curve into a linear equation that has been presented. The control system with a custom PCB has been developed to support all identified requirements for autonomous operation of the automated cab roof deflector. Figure 4-6 shows the main PCB with the mounted microcontroller AT90CAN128, the H-bridge interface on the right and the ADC input interface.

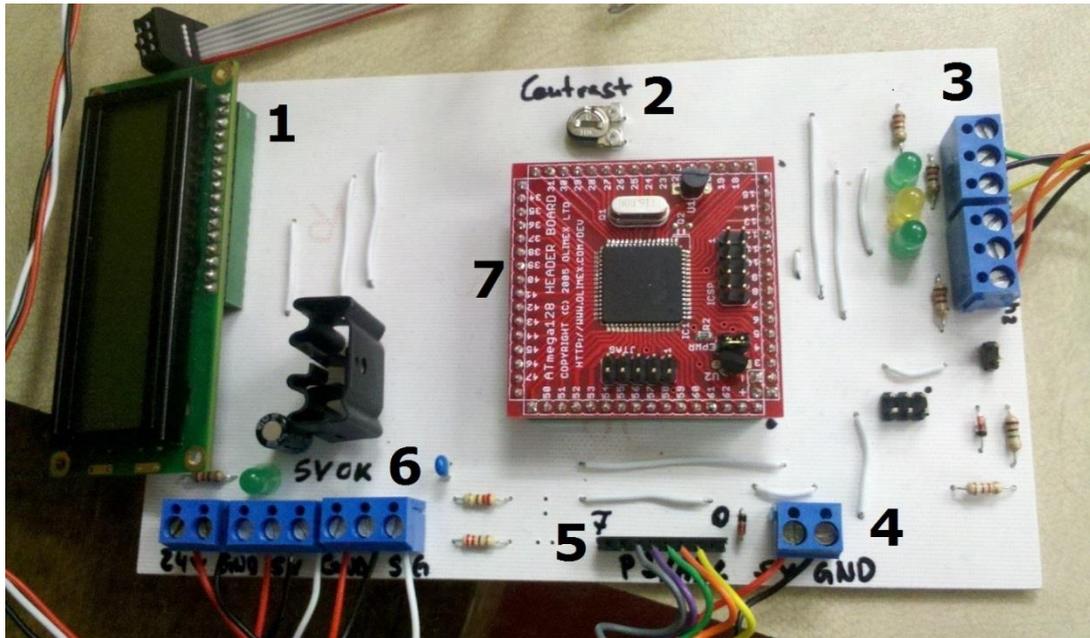


Figure 4-6: Picture of the developed main controller circuit board

The following components can be seen:

1. LCD Display for debugging
2. Potentiometer for display contrast adjustment
3. Outputs with indicator LEDs for controlling the H-Bridge (Figure 4-5)
4. 5V DC terminal for supplying the pressure sensors (Figure 3-7)
5. ADC input terminal connected with the outputs of the pressure sensors
6. IR Distance sensor terminals (Distance sensors positions can be seen in Figure 4-2 and Figure 4-3)
7. AT90CAN128 microcontroller board

Chapter 5

Software Development

The previous chapter presented the mechanical and electrical systems of the cab roof deflector to be automated. This included the mechanical dimensions of the deflector, the configuration of the sensor array as well as the selection of the sensors types. The minimum requirements for a microcontroller and the choice of an applicable microcontroller have been presented as well. A control circuit for the automation of the cab roof deflector system has been developed.

This chapter describes the development and the functionality of the control system software. This includes the implementation and evaluation of the used and already presented wind condition sensors and distance sensors. This section also includes the presentation about how Computational Fluid Dynamics (CFD) results are compiled to be suitable for implementation into time application based on an eight bit microcontroller and how the decision making has been built to adjust the cab roof deflector to optimum aerodynamic position.

5. Software development

The code for the ATMEL AT90CAN128 microcontroller unit (MCU) has been developed in ATMELs own development platform AVR Studio with version number 4.19-730 (2011) [34]. Also the free programmers' library WinAVR in version 20100110 has been used [35]. The library contains modules that facilitate programming by providing a number of commands that make use of microcontrollers' functions more accessible. The library modules also improve readability of the software code. The documentation on the WinAVR Library can be found online [36]. Figure 5-1 shows a screenshot of the AVR Studio development platform.

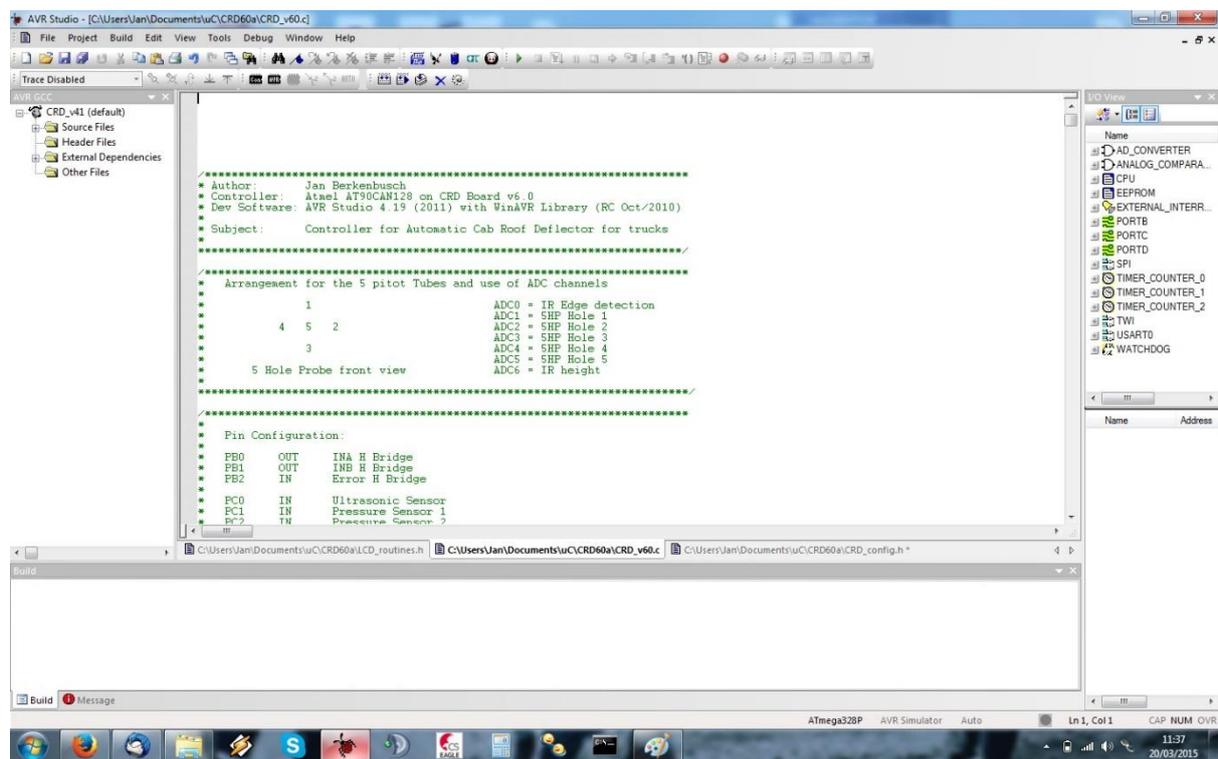


Figure 5-1: Screenshot of AVR Studio development platform

Figure 5-2 shows the USB to ISP interface Model mySmartUSB light that has been used to program the developed software into the microcontroller chip [37].



Figure 5-2: USB to ISP interface mySmartAVR lite

5.1 Software architecture

Embedded controller software always consists of three blocks:

- Controller configuration: The controller configuration consists the definition of input/outputs, the setup of timers, ADCs, communication interfaces, MCU clock source and other controller hardware related settings. It also includes the definition of used variables.
- System initialization: The system initialization consists of code and functions that has only to run once at the power-up of the system.
- Main programme endless loop: The main programme endless loop consists of the actual controller software, which is looped to fulfil a specific job. The main loop can be paused by interrupts to react to internal or external interrupt conditions.

5.1.1 Microcontroller configuration

The configuration in this application consists of the configuration of the controller clock source, of inputs and outputs, the configuration of the ADC and the configuration of Timer1.

The controller system clock source has been set to use the controllers' internal 8 MHz clock source. Though the use of the internal clock is not as precise as an external clock crystal it is easy to implement and reliable for applications that are not critical to time or timing.

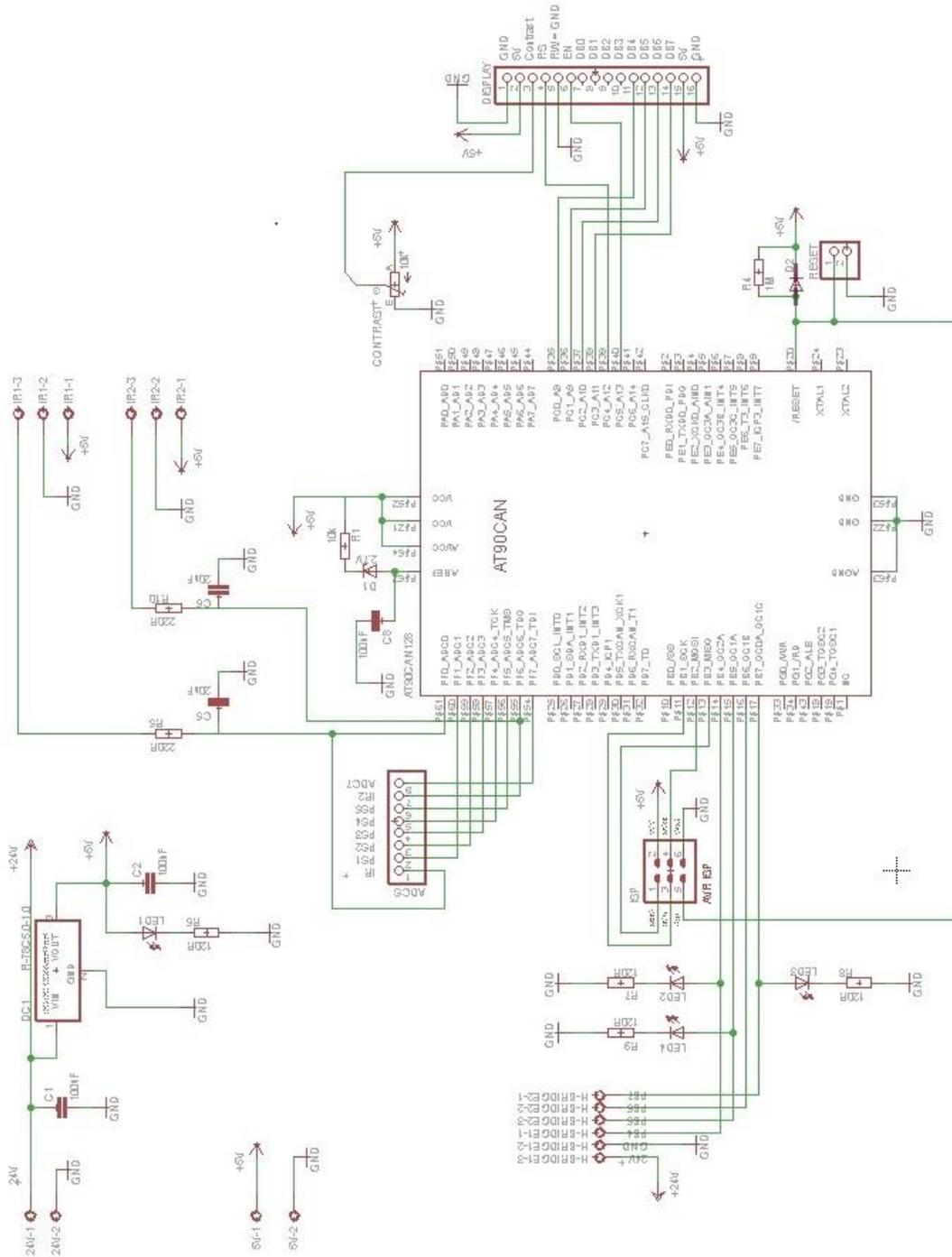


Figure 5-3: circuit schematic of the controller board

The input and output configuration is related to the design of the control circuit. In Figure 5-3 can be seen that the pin configuration depends on the devices connected to each output. The display for debugging is connected to Port C of the microcontroller. Therefore the Pins of Port C have been configured as outputs. The pin configuration for Port C can be found in Appendix A4 p. 122, 147 and 157. The motor is controlled with a high power H-bridge. Four pins of Port B are connected to the H-bridge and have been set as outputs. The other four pins of Port B are blocked as programming interface and therefore not used in the software. The pin configuration for the Port B can be found in Appendix A4 p. 122 and 124. The ADC functionality of Port F has been set up for acquisition of sensor outputs and are not used in binary input/output mode. The ADC functionality is activated by configuration of the ADC registers.

The ADC operate best at frequencies in a range of 50 KHz to 200 KHz [32 p. 276]. To set a ADC frequency in that range based on a system clock of 8 MHz a prescaler of 64 had to be used. This results in an ADC frequency of:

$$\frac{8 \text{ MHz}}{64} = 125 \text{ KHz} \quad (\text{Equation 9})$$

The ADC registers have been set to enable the ADC and to enable the “conversion completed” interrupt. Also the configuration registers have been set for an ADC prescaler of 64. The configuration of the sensor input pins of Port F can be found with the configuration of the ADC registers in Appendix A4 p. 124

The time intervals for updating the output signals of the sensors are 1 ms for the pressure sensors and 30 ms for the distance sensors respectively [25, 30]. This results in a minimum sampling time of 30 ms for the system. Since the system is not critical to time and with a slow response time of the chosen actuator the sampling time was chosen to be 500 ms. Timer1 has been used to create a sampling clock for the acquisition of sensor outputs. The timer registers have been set to “clear timer on compare match” mode with output compare register A. With this mode Timer1 counts up until the value of the timer register matches the predefined value of output compare register A. If both register values match the timer register is reset to zero and an interrupt is generated. The compare match register was set to 15625 and the timer prescaler for timer1 was set to 256, resulting in a timer clock of 500 ms (equation 10).

$$\frac{8 \text{ MHz}}{256} : 2 = \frac{8000000 \frac{1}{s}}{256} : 2 = 15625 \quad (\text{Equation 10})$$

The timer control registers have been set to enable mode “clear on compare match” CTC and to create an interrupt flag for fulfilled compare match condition. The setup of the Timer1 configuration registers can be found in Appendix A4 p. 124.

5.1.2 System initialization

The initialisation runs functions which are only run once each time the system is powered up. In this application the detection of the trailer reference height and the referencing of the five hole probe pressure sensors are executed. Once the trailer height is recognized this value is saved and for later use in the decision making function. This function only has to be executed at each system start-up, since the trailer height will remain unchanged during its journey.

The deflector shield with attached proximity sensor (Figure 4-2 and 4-3) is raised until the sensor signal exceeds a threshold value which represents that the sensors is above the trailer roof level. Afterwards the deflector shield is lowered until the proximity sensor crosses the threshold again. As the sensor finds the trailer top edge the deflector movement is stopped and the deflector position is saved as trailer reference height. The deflectors’ position is measured by determination of the trailer top edge position. The height difference of 115 mm between the deflector top edge and the position of the proximity sensor is respected in the trailer reference height. The source code for the trailer height detection can be found in Appendix A4 p. 125.

The other initialisation step is to reference the pressure sensors connected to the five hole probe. To do so the sensor offsets are calculated for each sensor separately. The resolution of the ADCs in the chosen microcontroller is 10 bits or 1023 increments. The ADC reference voltage is divided by 1023 increments hence one increment is equal to:

$$3 \text{ V} : 1023 = 3000 \text{ mV} : 1023 \approx 2,93 \text{ mV} \quad (\text{Equation 11})$$

If the ADC output value exceeds 1023 the input voltage is at least equal to the ADC reference voltage of 3 V DC. As mentioned in chapter 3.5 the reference voltage for zero differential pressure vary slightly at about 2.26 V DC for all pressure sensors P_1 to P_5 , which is equal to an ADC value of:

$$2260 \text{ mV} : 2,93 \text{ mV} = 771 \quad (\text{Equation 12})$$

In chapter 3.5.1 the condition of zero differential pressure has been defined as atmospheric pressure of 101300 Pa. Figure 3-14 shows that the maximum of measured pressures is at 101300 Pa. Hence it has been concluded that the ADC values do not exceed the reference value

during system operation but are expected to be lower than the reference condition. Because of the varying different reference output voltages of the pressure sensors it was chosen to even out the ADC reference values. The ADC reference values have been evened out with software offsets. The ADC reference value has been chosen considerably lower than the maximum of 1023, but above the highest ADC value that was read without offsets involved. The reference ADC value of 900 has been chosen to make display outputs of ADC values for debugging as clear as possible. The reference value itself is not important to the operation of the system since the algorithm operates on differentials of ADC values. To set the ADC reference to 900 independent from the environmental influences on the sensor output the offset for each pressure sensor $Offset_{PS_n}$ is calculated individually during the system start up. This has been implemented by calculating each pressure sensor offset as

$$Offset_{PS_n} = 900 - ADC_{PS_n} + 1 \quad (\text{Equation 13})$$

The source code for referencing the ADCs can be found in Appendix A4 p. 125.

5.1.3 Main programme

The main routine runs in an endless loop and only covers the tasks that have to be run during the trucks journey. The tasks are: reading sensor output values, calculation of wind conditions, calculation of the deflectors' optimum positioning and adjustment of the deflector position.

Figure 5-4 shows the schematics of the main routine of the control system. The development of the controller software has been following the Input-Processing-Output principle known as IPO. The system reads the outputs of five pressure sensors and two IR sensors and converts the readings in pressures and position information respectively. The wind information is calculated from the pressure sensor values. The optimum position for the deflector is calculated from the trailer reference height and the wind condition information. Based on the deflector optimum position and the current deflector position the decision is made whether the deflectors' position has to be adjusted.

The acquisition of the sensor output values is achieved with interrupt commands.

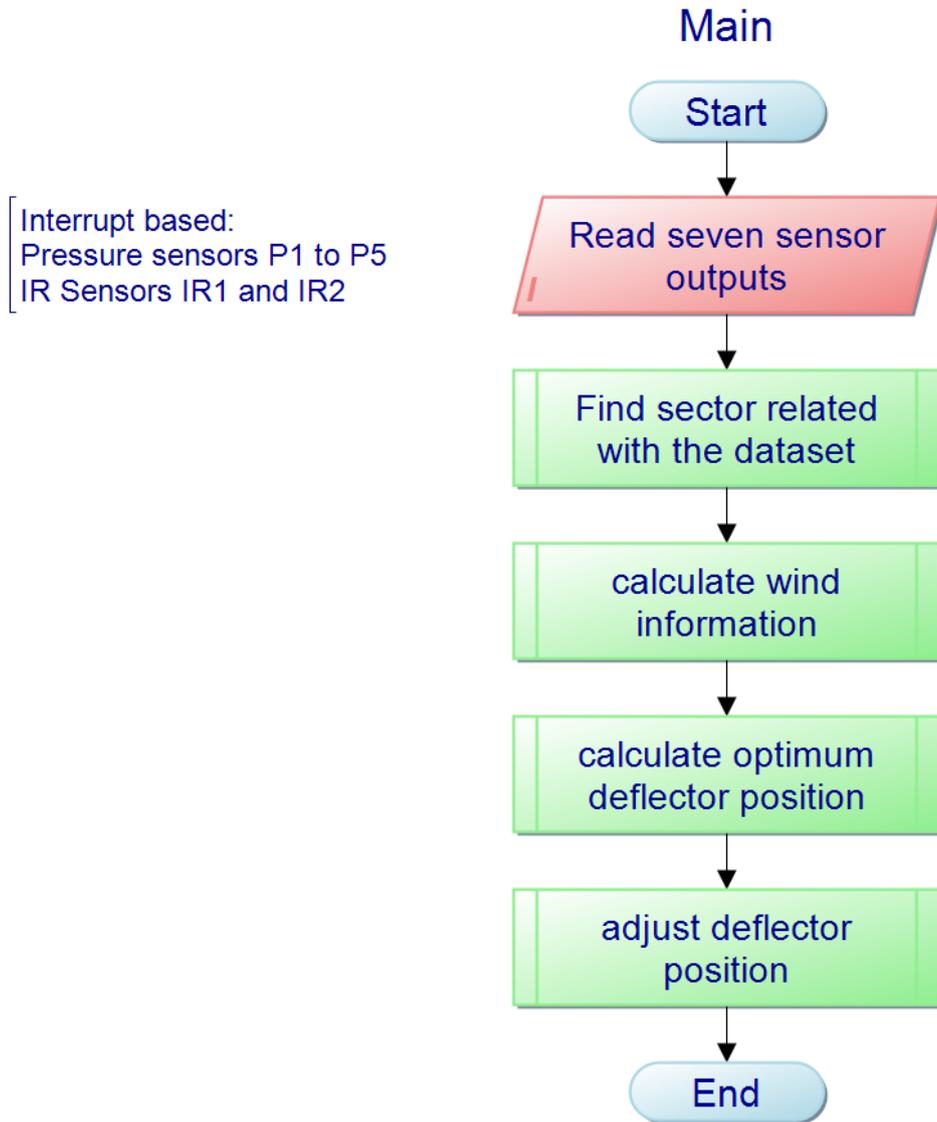


Figure 5-4: Controller software main routine flowchart

5.1.3.1 Interrupt structure

To achieve an acquisition of sensor output values in equal intervals timer based interrupts have been set up to control sensor output acquisition. Interrupts are small pieces of software that respond to certain internal or external triggers. Interrupts pause the main software and execute themselves. To cause minimum delay in the main programme it is crucial that the execution time for the code of an interrupt is short as possible. After an interrupt has finished executing its code the main routine code is continued. If an interrupt trigger comes up by the time an

interrupt is already active the 2nd interrupt flag is saved and executed after the 1st interrupt is finished. Because microcontrollers cannot read input levels exactly at the same time the feature of stacking interrupt triggers was used to cascade sensor reading triggers. That software design allows to read input signals on ADCs as simultaneously as possible with only a minor delay in μs range between two readings [32]. The sensor output values are read into the controller. They are converted into values of pressures and distances respectively by applying the sensor output curves provided in the particular datasheets [25, 30]

The controllers' internal clock signal created by timer1 has been used to trigger the acquisition of sensor inputs. This has been done with the "timer compare match" interrupt setting the "start conversion flag" for the ADC. The ADC has been configured to set an "ADC conversion completed" flag at the end of a successful ADC conversion (see Chapter 5.1.1). That flag triggers the "conversion completed" interrupt. The "conversion completed" interrupt code is executed. That interrupt copies the values from the ADC data register into variables for further processing and automatically clears the ADC data register. Because eight ADC channels are multiplexed on the same actual ADC the ADC multiplexing channel has to be selected for each conversion. To do so a channel selection variable has been used in combination with a "switch-case" statement. Each code for a certain case follows the same structure, with slight variances between the distance sensor and pressure sensor cases. The "conversion complete" interrupt consists of seven cases to save the sensor output correctly in the related sensor value variable. The data acquisition code for a single case has been programmed as follows:

- The data in the ADC data register is copied to an array variable with five elements each.
- The values for each sensor are averaged over the array of five elements and copied to a sensor output variable that is used for calculations.
- The channel selection variable and channel selection register are increased
- The "start ADC conversion" flag is set to start the next ADC conversion immediately after the current interrupt is finished.
- The end of the seventh case in the interrupt is the end of the acquisition of a sensor output dataset. The channel selection variable and the channel selection register are reset. The "start ADC conversion" flag is not set, but another flag that activates the further processing of sensor output values into wind condition information. The array

counter is increased after the acquisition of the 7th sensor and reset if the counter reaches the upper array limit.

The source code for “conversion completed” interrupt can be found in Appendix A4 p. 126.

5.1.3.2 Wind information calculation

The realized five hole Probe calibration method by A. R. Paul et al. [26] is based on sector method and a 4th order regression leading to a number of calibration coefficients (see Chapter 3.5). Wind information consisting of velocity v , pitch angle α and yaw angle β can be calculated with a set of five hole probe pressure readings and the full set of calibration parameters (see Appendix A1). The process of calculating wind information from pressure values obtained by a calibrated five hole probe can be found in chapter 3.6. This chapter describes how the process was adapted into source code.

To identify the sector of a set of pressure readings the highest value within a set of pressure readings of P_1 to P_5 has to be determined. Therefore the value of pressure P_1 is copied to variable `highest` and the value of variable `sector` was set to 1. The value of P_2 was compared with the value of `highest`. If P_2 was greater than `highest` the value of `highest` was updated to the value of P_2 and the variable `sector` was updated to 2. This process was performed for all pressures P_2 to P_5 . The process concludes with the highest absolute pressure copied to variable `highest` and the sector of the dataset stored in the variable `sector`. This process is shown as a flowchart in Figure 5-5.

As mentioned in chapter 3.5 and 3.6 the processing of pressure values depends on how the five hole probe tip is positioned to the stream to be measured. The positioning of the probe in the stream is defined by the sector model. The value of variable `sector` is used for selecting the correct set of calculations and calibration parameters in further processing. The source code for the sector determination can be found in the bottom of the page of Appendix A4 p. 127.

Find sector related with the dataset

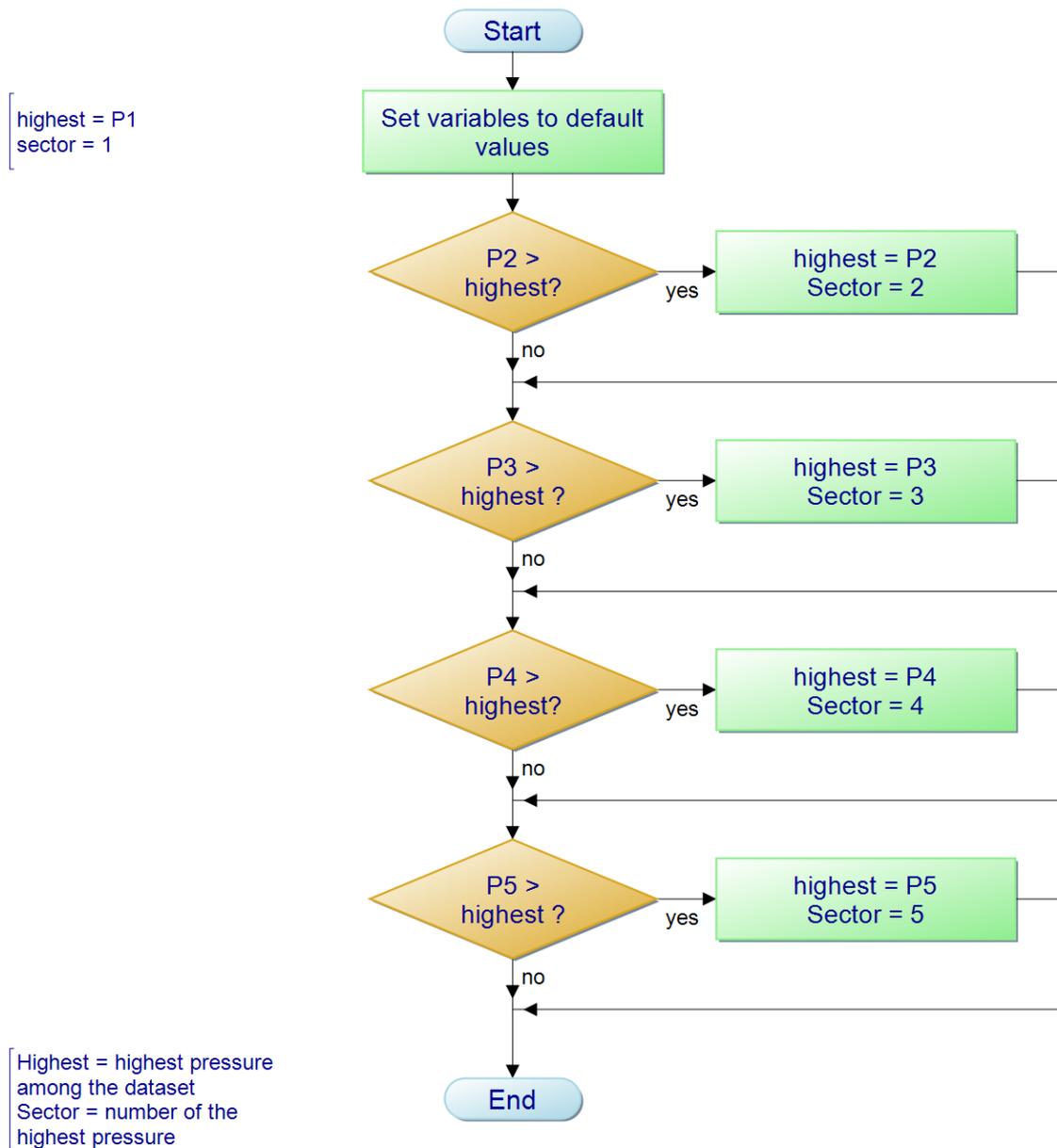


Figure 5-5: Controller software for finding the sector of the sensors dataset flowchart

Figure 5-6 shows the flowchart of the implemented switch-case statement that uses the value of `sector` as argument to calculate full three dimensional wind information. For each sector the corresponding calibration parameters and equations were implemented to correctly calculate the wind information (see Chapter 3.6). Each case follows the same structure for the calculations: The pressure coefficients CP_{α} ($= C_{p_{\alpha}}$) and CP_{β} ($= C_{p_{\beta}}$) are calculated directly from the pressures P1 to P5 (equation 6). CP_{α} and CP_{β} are

processed into wind vector angles alpha ($= \alpha$) and beta ($= \beta$) as well as C_{Ptotal} ($= C_{Ptotal}$) and $C_{Pstatic}$ ($= C_{Pstatic}$) (equation 7). The pressure coefficients C_{Ptotal} and $C_{Pstatic}$ are processed into actual pressure values P_{total} ($= P_{total}$) and P_{static} ($= P_{static}$) by application of equation 8. The wind velocity v ($= v$) then is calculated with equation 3 (see Chapter 3.1). The source code for the wind vector calculation can be found in Appendix A4 p. 129-143.

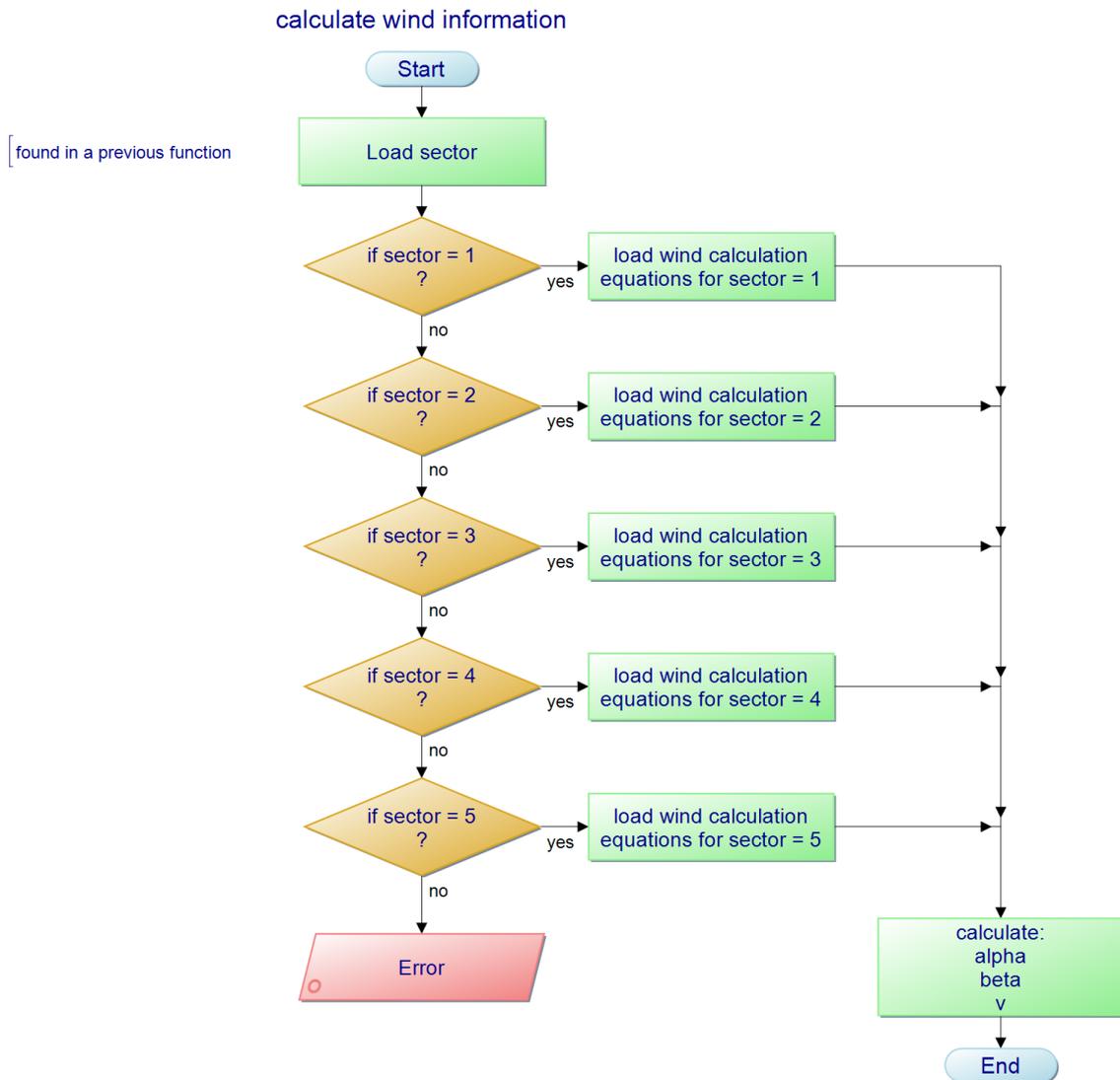


Figure 5-6: Controller software wind information calculation flowchart

The resulting three dimensional information consisting of α , β and v in the wind information vector is used for the calculation of the optimum deflector position.

5.1.3.3 Deflector height related calculations

The programmable model for the optimum positioning of a cab roof deflector was developed based on simulations in computational fluid dynamics (CFD). The simulations were set-up and performed by PhD Mechanical Engineering Research Student Mr. Isuru Sendanayake Achchige (Isuru.Sendanayake@hud.ac.uk).

A number of simulations with various mesh elements were carried out for mesh independence testing. The number of mesh elements ranged from 0.25 million elements to 12 million elements. It was found that increasing the number of mesh elements from 8 million to 10 million mesh elements only resulted in a deviation in the drag force of less than 1 % between both simulations (Figure 5-7). Therefore it was concluded that a mesh of 8 million elements is sufficient for simulating drag force results on a truck model. All simulations were performed with a mesh of 8 million elements.

The initial set of performed simulations features the geometry of a given deflector (Figure 4-1) and various trailer height, various wind velocities and various yaw angles of the flow.

The results of the initial simulations that have been used can be found on DVD in the appendix A5.

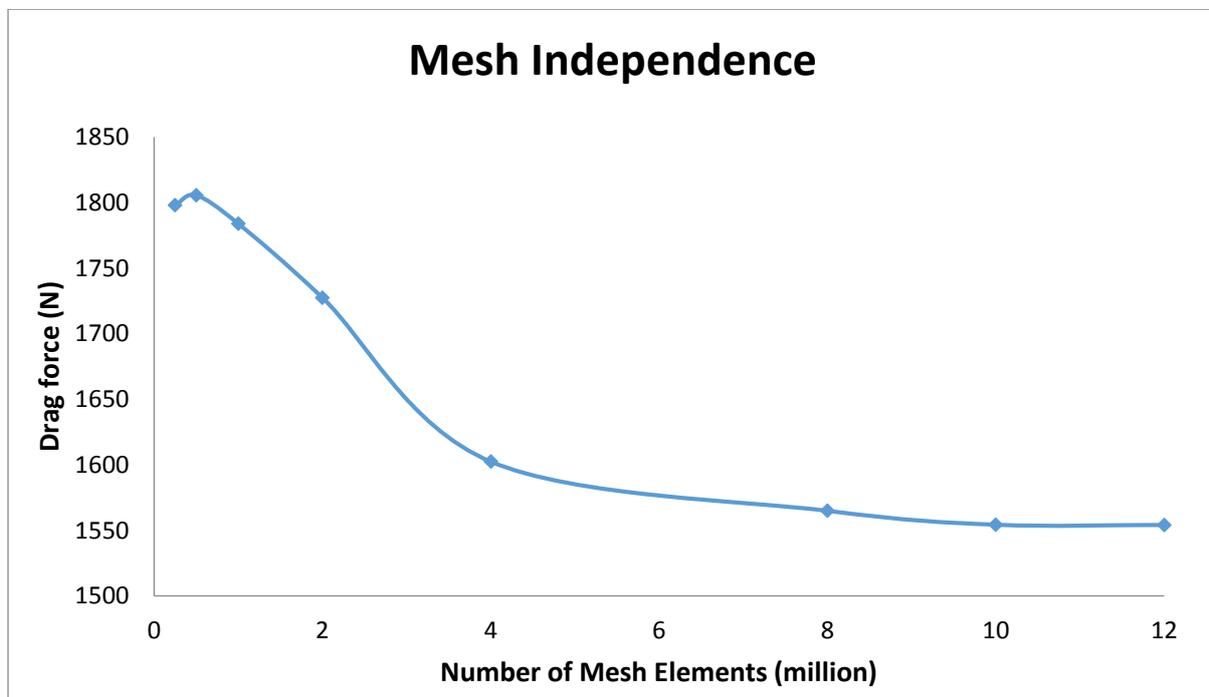


Figure 5-7: Mesh independence simulation results

To find a model for the optimum deflector position a computational model for CFD simulations has been created. The model consisted of an articulated truck with attached CRD. The simulated CRD has been modelled to match the actual CRD used in this project and presented in chapter 4.

To find the optimum deflector positioning numerous simulations with combinations changing wind velocities, changing crosswind angles, changing deflector reference height and changing deflector positioning have been evaluated. The wind velocities have been defined as 40 mph, 50 mph and 56 mph. The chosen yaw angles (respecting the symmetry of the articulated vehicle) have been defined as 0 °, 10 °, 20 ° and 30 °. The trailer reference height has been defined as 4.2 m, 4.6 m and 4.8 m. The deflector positions have been chosen with reference to the deflector top edge on the same height. The deflector top edge has been chosen to deviate within ± 200 mm from that reference position in 100 mm increments. Combinations of all these adjustments led to 180 simulations that have been evaluated. The CFD results can be found on DVD in Appendix A5.

To match the limited computing power of the microcontroller the CFD results have been compiled into a mathematical model that represents the optimum deflector position and is applicable for microcontrollers.

The CFD results for combinations of yaw angle, wind velocity, trailer reference height and deflector positioning have been transferred into an excel sheet to give an overview on all CFD results. The transferred data has been evaluated for the least drag among the sets with the same trailer reference height, velocity, yaw angle and only changing deflector position. The relative vehicle speed also includes crosswind effects, as stated in chapter 3 and figure 3-3. The resulting combinations of reference trailer height, yaw angle and vehicle speed that lead to least drag force on the HCV are shown in Table 5-1.

4.2m		yaw			
		0	10	20	30
40		4.4	4.4	4.4	4.4
50		4.4	4.4	4.4	4.4
56		4.4	4.4	4.4	4.4
mph					

4.6m		yaw			
		0	10	20	30
40		4.5	4.7	4.8	4.8
50		4.5	4.7	4.8	4.8
56		4.5	4.7	4.8	4.8
mph					

4.8m		yaw			
		0	10	20	30
40		4.6	4.7	4.9	4.9
50		4.6	4.7	4.9	4.9
56		4.6	4.7	4.9	4.9
mph					

Table 5-1: Deflector heights from CFD results for aerodynamically optimum positioning in m

For a better visualization the optimum deflector position with respect to the reference trailer height is coloured. Green values represent an optimum deflector position that is lower than the trailer reference height, yellow values represent a deflector position of ± 0 mm to the trailer reference height, orange values represent a deflector position higher than the trailer reference height.

Aerodynamic results for negative yaw angles have not been simulated since the vehicle is symmetrical and the results for positive and negative yaw angles will result in the same values.

For visualisation of the further processing the tables for the deflectors' optimum position have been plotted as three dimensional surfaces. Pitch and yaw angles can be found as inputs on the X and Y axis, the deflectors' optimum position as output can be found on the Z axis respectively.

Figure 5-8 shows the positioning surface for the deflector in case of a 4.2 m trailer reference height. It can be seen, that the optimum deflector positioning is not changing and for any combination of speed and yaw angle. The resulting optimum deflector position is always at 4.4 m for a trailer reference height of 4.2 m.

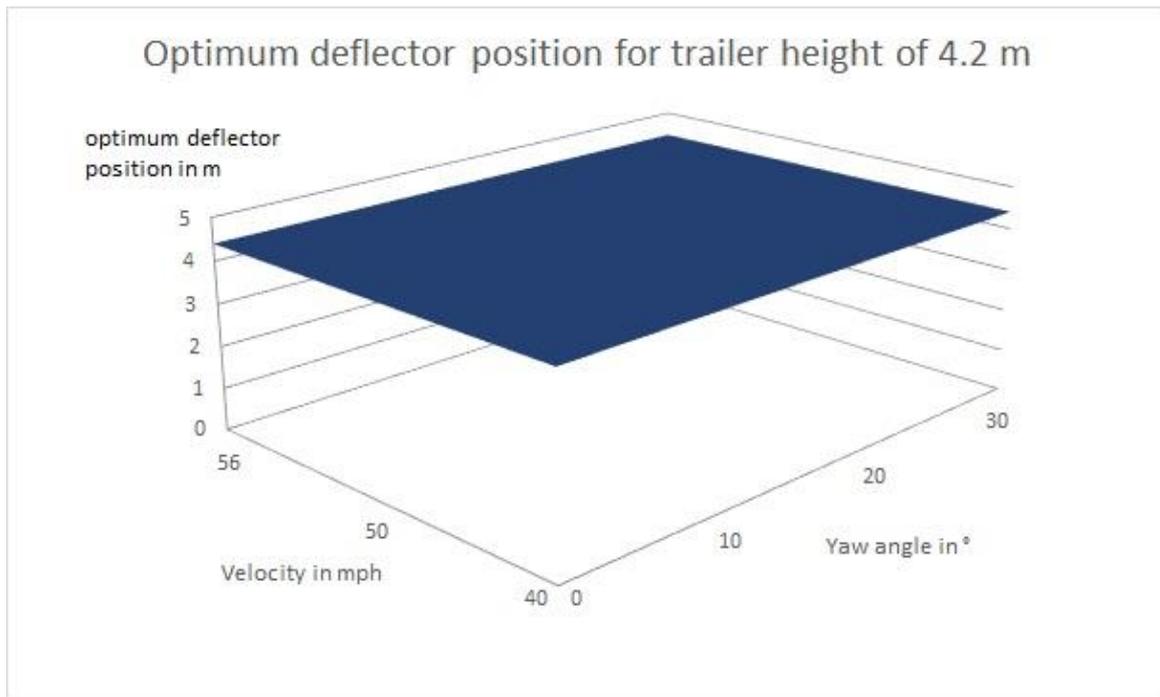


Figure 5-8: Surface representing optimum deflector positioning for 4.2 m reference height

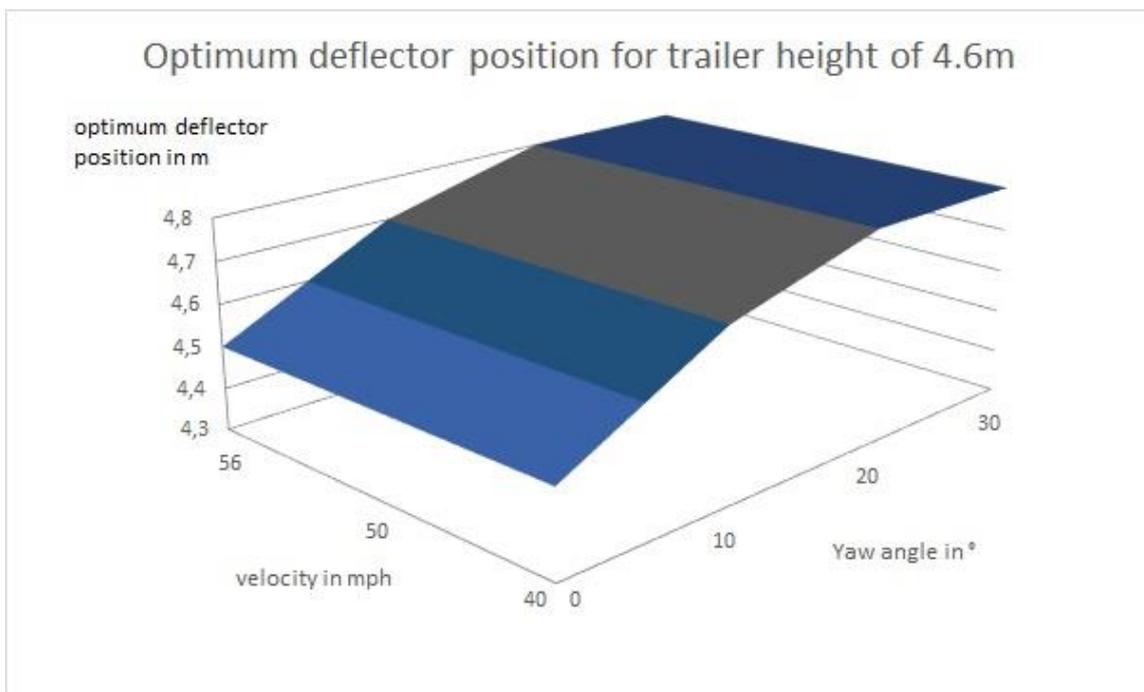


Figure 5-9: Surface representing optimum deflector positioning for 4.6 m reference height

Figure 5-9 shows the optimum deflector positioning for a trailer with a reference height of 4.6 m. It can be seen, that the greater the yaw angle becomes, the higher the deflector has to be

set. The wind speed does not have an impact on the deflector positioning, but the slope of the surface is changing with changing yaw angles.

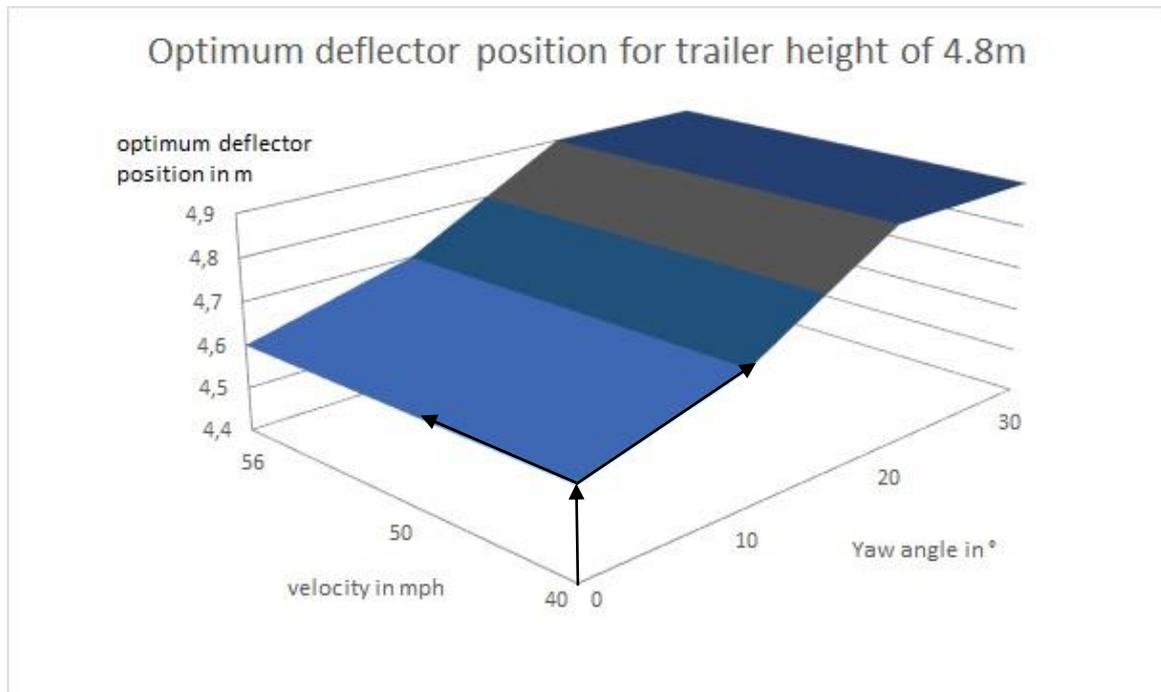


Figure 5-10: Surface representing optimum deflector positioning for 4.8 m reference height

Figure 5-10 shows the optimum position of the deflector with respect to a trailer reference height of 4.8 m. It can be seen that the optimum position of the deflector is independent from the vehicle speed, but dependent on the crosswinds yaw angle. The slope of the surface is changing as it appeared in Figure 5-8, too.

To describe the surface with changing slopes in mathematical equations the surface has been split in different sections forming one equation per section with a steady slope. An example for the further processing is given for Figure 5-10 and the case of a 4.8 m trailer height.

In Figure 5-10 the surface section for velocity 40 mph to 56 mph and yaw angle 0 ° to 10 ° shows a resulting optimum deflector position of 4.6 m to 4.7 m. That surface section is treated as a three dimensional plane and described with vectors. The plane is evolved around the point $(\beta|v|h) = (0|40|4.6)$. The direction vectors follow the plane section in velocity direction and in yaw angle direction. The point and the direction vectors are shown in Figure 5-10. The same vectors are inserted into a section of table 5-1, too. That section with the vectors inserted is shown in Figure 5-10.

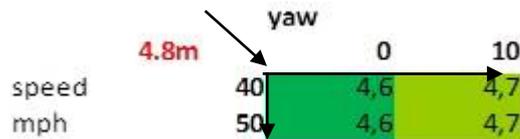


Figure 5-11: Spots of positioning surface for 4.8 m sector 1

Point $(\beta|v|h) = (0|40|4.6)$ and both direction vectors form a vector equation that is shown in equation 12 and equation 13:

$$(\beta|v|h) = (0|40|4.6) + s(10 - 0|40 - 40|4.7 - 4.6) + t(0 - 0|50 - 40|4.6 - 4.6) \text{ (equation 14)}$$

$$\Leftrightarrow (\beta|v|h) = (0|40|4.6) + s(10|0|0.1) + t(0|10|0) \text{ (equation 15)}$$

Equation 13 is equal to the system of linear equations shown in Figure 5-11:

		s	t
yaw	0	10	0
speed	40	0	10
height	4,6	0,1	0

Figure 5-12: Equation system for deflector optimum height 4.8 m section

The linear equations of Figure 5-12 have been solved for the deflector height h . This results in an equation of height h which is dependent on input variables yaw angle β and velocity v (equation 14). Equation 14 is applicable for trailer reference height 4.8 m and any combination of beta and v with $0^\circ < \text{yaw angle } \beta < 10^\circ$ and $40 \text{ mph} < \text{wind velocity } v < 50 \text{ mph}$:

$$h(\beta, v) = 0.01\beta + 4.6 \text{ (Equation 16)}$$

That process has been applied for all datasets of optimum positioning tables. The first sector was defined at the top left of each table and sector six in the bottom right of the table with each sector covering a set of four points (Figure 5-13).

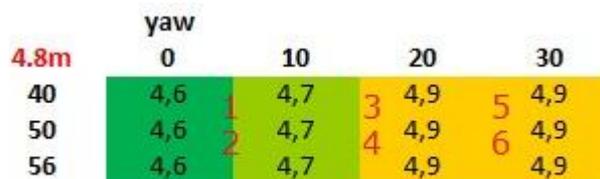


Figure 5-13: Nomenclature of sectors in the positioning table for 4.8 m trailer height

All deflector optimum positions obtained from CFD simulation (Table 5-1) have been processed the same way. The resulting sets of equations that exactly represent the deflector positioning surfaces (Figure 5-8, Figure 5-9, Figure 5-10) are shown below:

4.8 m	0-10 °	10-20 °	20-30 °
40-50 mph	$h(\beta, v) = 0.01\beta + 4.6$	$h(\beta, v) = 0.02\beta + 4.5$	$h(\beta, v) = 4.9$
50-56 mph	$h(\beta, v) = 0.01\beta + 4.6$	$h(\beta, v) = 0.02\beta + 4.5$	$h(\beta, v) = 4.9$

Table 5-2: Equations for optimum deflector height for 4.8 m trailer height

Table 5-2 contains equations that describe the deflector optimum position for a trailer reference height of 4.8 m (Figure 5-9). It can be seen that the characteristic gain of the middle part for yaw angles between 10 ° and 20 ° is twice as much as the gain in the part of yaw angle between 0 and 10 °.

4.6 m	0-10 °	10-20 °	20-30 °
40-50 mph	$h(\beta, v) = 0.02\beta + 4.5$	$h(\beta, v) = \beta + 4.7$	$h(\beta, v) = 4.7$
50-56 mph	$h(\beta, v) = 0.02\beta + 4.5$	$h(\beta, v) = \beta + 4.7$	$h(\beta, v) = 4.7$

Table 5-3: Equations for optimum deflector height for 4.6 m trailer height

Table 5-3 contains equations that describe the deflector optimum position calculation for a trailer reference height of 4.6 m (Figure 5-8). It can be seen that the characteristic gain of the first part for yaw angles between 0 and 10 ° is twice as much as the gain in the middle part of yaw angle between 10 and 20 °.

4.2 m	0-10 °	10-20 °	20-30 °
40-50 mph	$h(\beta, v) = 4.4$	$h(\beta, v) = 4.4$	$h(\beta, v) = 4.4$
50-56 mph	$h(\beta, v) = 4.4$	$h(\beta, v) = 4.4$	$h(\beta, v) = 4.4$

Table 5-4: Equations for optimum deflector height for 4.2 m trailer height

Table 5-4 contains equations that describe the deflector optimum position calculation for a trailer reference height of 4.2 m (Figure 5-7). It can be seen that the surface is flat and horizontal, so there is no gain factor in the equations.

Because of the symmetry of the articulated trucks all simulations have only been run for positive yaw angles. Negative yaw angles are multiplied in the software by -1 to represent that symmetry. Equations for calculation of the deflectors' optimum position have only been developed for positive yaw angles.

Figure 5-14 shows a flowchart on how the optimum position equations obtained from the CFD results have been implemented in the software code. The programmable equations for optimum deflector positioning (Table 5-2, Table 5-3, Table 5-4) have been implemented in an equation selection system. The equation selection has been built with nested if-statements that chose the deflector positioning equation based on measured wind information and measured trailer reference height. The yaw angle is checked if it is below zero. If the yaw angle is negative it gets multiplied by -1. Secondly the reference trailer height is checked. In case the reference trailer height is not recognized and there is no trailer mounted the system skips the deflector optimum position calculation. If a trailer height is recognized, for example the 4.2 m, the system follows that tree. The system checks for the yaw angle next. It loads the equation for the certain trailer height and yaw angle situation and calculates the trailer optimum position. The source code for the equation selection system can be found in Appendix A4 p. 144 and 156.

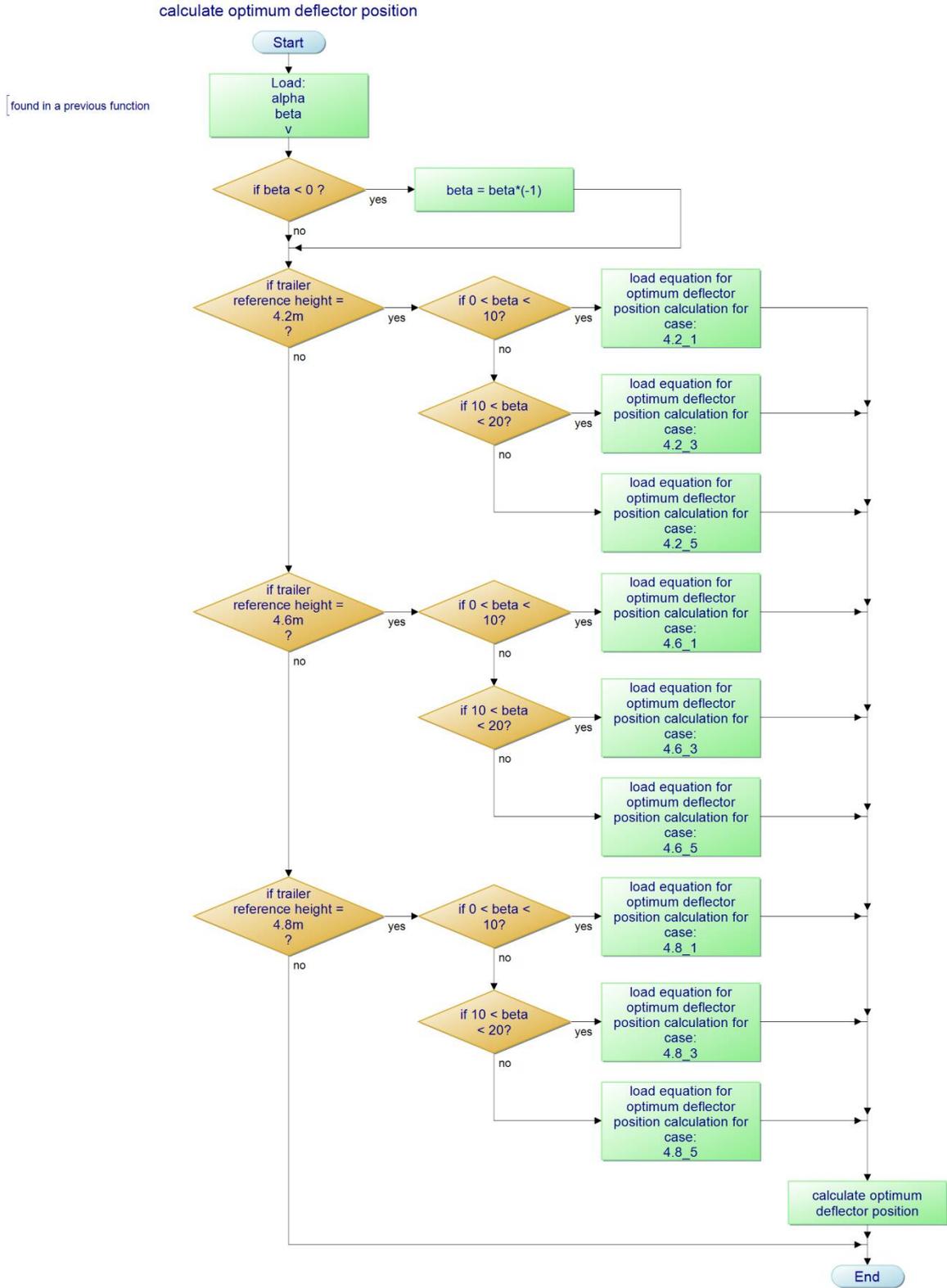


Figure 5-14: Controller software optimum deflector position calculation flowchart

The decision whether to adjust the deflectors position is based on the calculated deflectors optimum position and the deflectors current position.

5.1.3.4 Deflector positioning

With known optimum position for the CRD the deflector needs to be adjusted to that position. During the process of adjustment it is inevitable that the deflector positioning gets checked frequently and that the current deflector positioning is updated. Therefore it has been chosen to build the adjustment function as a conditional switching of the actuator with three states of operation: The states are: moving up, moving down and stopped movement.

In combination with the interrupt based sensor output acquisition the system can recalculate the optimum deflector during the adjustment process. It also can respond to changing optimum position in real time.

The software calculates the positioning difference Δh between the actual deflector position and the optimum deflector position. The decision whether to move the deflector up, move the deflector down or to stop the deflector is made if the Δh is over or below certain threshold values. Figure 5-15 shows a flowchart of how the deflector positioning was implemented. First in this process the optimum deflector position is loaded. Secondly the difference Δh between trailer height and trailer optimum position is calculated. If Δh is lower than -3 cm the outputs are set to move the deflector up. If Δh is higher than +3 cm the outputs are set to move the deflector down. If Δh is between -1 cm and +1 cm the outputs for moving the deflector are reset. This results in a function that adjusts the deflector if Δh is higher than ± 3 cm to a value of Δh of less than ± 1 cm. The chosen setup stops the deflector movement if Δh is smaller than ± 1 cm (which compensates some motor overrun during the switching off process) and starts the adjustment process when Δh is greater than ± 3 cm. The source code for the deflector positioning can be found at Appendix A4 p. 145

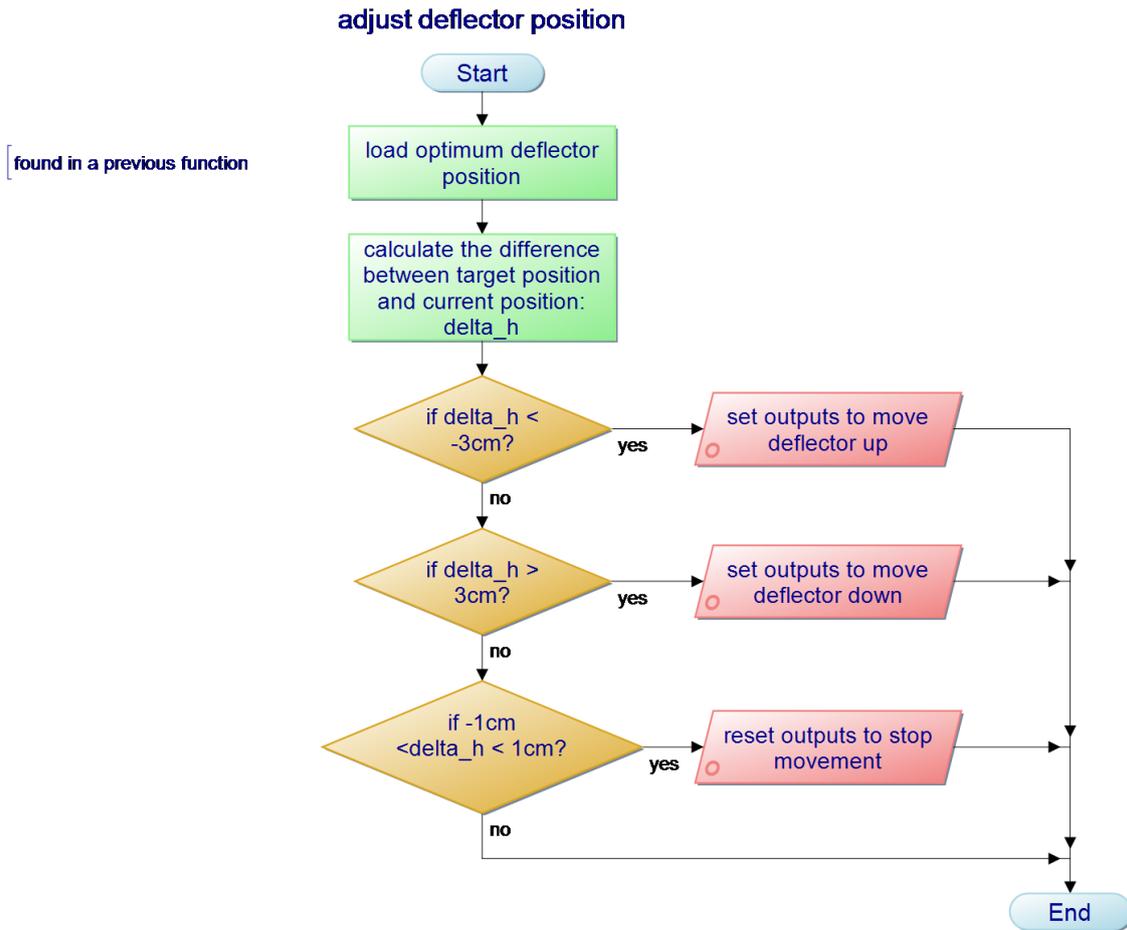


Figure 5-15: Controller software deflector position adjustment flowchart

Application of the mentioned software structure required defined threshold values for whether the deflectors' current position is sufficient with respect to the calculated optimum position. Tests have shown that a too tight positioning threshold leads to oscillating deflector shield behaviour, which is caused by fluctuation in the sensor outputs. A too wide threshold for the deflector positioning reduces the effectiveness of the automated CRD because then the CRD will be too far off the optimum position to have a positive aerodynamic effect.

5.2 Conclusion

Chapter 5 presented the software structure for the controller software, which was developed for this research project. The configuration of the timer and interrupt based data acquisition procedure was presented. This includes the cascading algorithm to acquire sensor readings as simultaneously as possible and how the sensor value related calculations have been realized.

The compilation of CFD results into computable aerodynamics related equations that are suitable for little computing power such as microcontrollers was presented. The CFD results have been sorted by the best aerodynamic result and the least drag force. The combination of aerodynamic best deflector position and related wind conditions have been compiled into vector equations. These equations have been solved for the variable “optimum position” and implemented with an equation selection function based on wind conditions to choose the correct equation.

The decision whether the deflector position needs adjustment is based on the deflector optimum position and the deflectors’ current position. To do so the difference between the optimum deflector position and the current deflector position is calculated. If the positioning difference exceeds a threshold of 3 cm the deflector position is changed. If the position difference is less than 3 cm the deflector position is not changed to save electrical energy and to avoid oscillation of the deflector due to sensor output fluctuation.

Chapter 6

Performance evaluation of the developed system

The previous chapter mainly described the design and the engineering process of the control system software for an automated cab roof deflector and related sensor components.

This chapter describes how the system was tested for function and analyses the test results for suitability of the application.

Bridge interfaces the control signals sent by the MCU with power circuit and actuator. The actuator moves the deflector shield and adjusts it to optimum positioning.

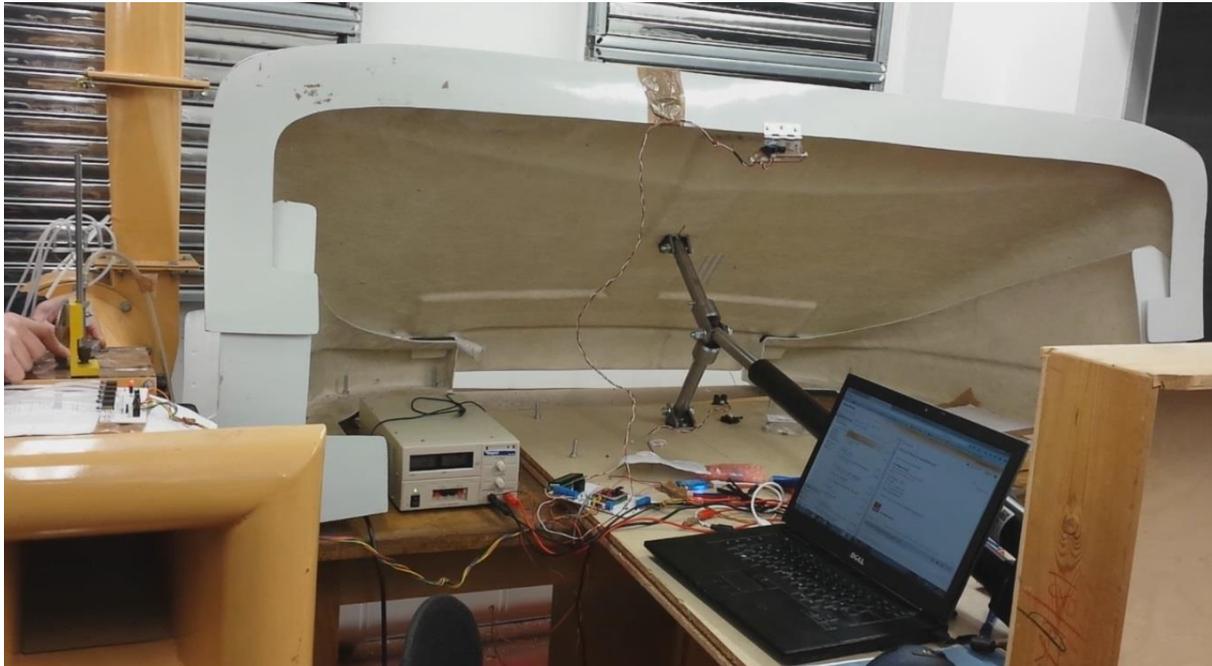


Figure 6-2: Test setup of the automated cab roof deflector system

Figure 6-2 shows an overview with all components of the system to be tested. The five hole probe tip has been mounted in the wind tunnel. The appendant pressure sensors have been mounted on the flow domain. The tip of the five hole probe and the pressure sensors have been connected to each other with rubber tube. The five hole probe, the pressure sensors and the tube can be seen on the left. The IR sensor for the trailer detection has been mounted on a hinge on the top of the deflector shield. The IR sensor for measuring the deflector position is oriented vertically and has been screwed to the table. It can be found just under the interfacing mechanic between actuator and deflector shield. All seven sensors have been connected to the circuit board with coloured signal wires.

The control circuit has been powered by an external power supply. The control system can be seen in Figure 6-2 on the table between the power supply and the laptop. Half of the actuator is hidden behind the laptop screen. The H-Bridge appendant with the actuator is also hidden behind the laptop screen. The vertically mounted wooden board used for the simulation of the trailer front surface can be seen on the bottom right of Figure 6-2.

The test has been designed to prove functionality of the developed system and if it covers the defined research aims. It has been set up to test all decisions made in the software to prove full

functionality. The five hole probe has been disposed to changing wind conditions to test the measurement component of the software. The trailer reference height has been changed to test the equation selection component of the software. The deflector positioning has been used as evaluation variable to test the deflector positioning component of the system. This test setup ascertains that evaluation of the deflector positioning gives insight if the system functions as designed.

The deflector position is primarily dependent on yaw angle β and the trailer height. The effect of changing wind velocity relative to the HCV is limited (Table 5-1). The equations that have been developed from the CFD results and have been implemented with the system do not use the flow velocity as input variable for calculating the optimum deflector position (Table 5-2 to Table 5-4). Hence it was chosen to test the system with changing yaw angles and changing reference trailer heights only. Varying flow velocities have not been tested.

The changing wind conditions have been simulated by mounting and turning the five hole probe in a wind tunnel, as it was previously done for the calibration process. The wind condition yaw angles to be evaluated have been chosen as 0° to 30° with increments of 5° . The trailer reference heights have been chosen as 4.2 m, 4.6 m and 4.8 m. Four values have been noted down for the evaluation of each combination of these conditions:

- The calculated deflector position that is used as reference value. These values are obtained from Table 5-2, Table 5-3 and Table 5-4 respectively.
- The deflector position that is measured (and displayed) by the system.
- The deflector position that is calculated as target by the system.
- The deflector position that was measured with a tape measure

Due to mechanical limitations of the deflector shield components it has been necessary to implement positioning limitations to not damage any deflector component during automatic operation. The highest possible deflector position without damaging any component is 78 cm above the table plate. The lowest possible deflector position possible is 62 cm above the table. These limitations enable a maximum deflector movement of 16 cm up and down depending on the deflectors' reference position. Positioning values that exceed these limitations are overwritten with the maximum feasible positioning values. The mechanical limitations required editing of the trailer reference height in the MCU software to test the control system for application with different trailer reference heights. The mechanical limitations also required

that the reference position was changed between both the highest and the lowest deflector position. This was necessary to test adjustment with target positions both higher and lower than the trailer reference height.

The reference values named **Calculated** have been calculated by inserting the yaw angles in the deflector positioning calculations shown in Table 5-2 to Table 5-4. Values labeled as **System** represent the values that the system measures for the one particular deflector position. Values labeled **Target** represent the calculated optimum positions for the deflector target position. These can be overwritten by the software end switch. Values named as **Measured** represent deflector positions that have been measured by hand.

Yaw angle β	0 °	5 °	10 °	15 °	20 °	25 °	30 °
Calculated	4.6	4.65	4.7	4.8	4.9	4.9	4.9
System	4.61	4.68	4.74	4.79	4.79	4.79	4.79
Target	4.62	4.66	4.72	4.78	4.78	4.78	4.78
Measured	4.62	4.70	4.76	4.81	4.81	4.81	4.81

Table 6-1: Test results for 4.8 m for changing wind angles (in m)

Table 6-1 shows the results of the test procedure for trailer reference height of 4.8 m. The upper limit with a cab roof height offset of 400 cm is 4.78 m and the lower limit is 4.59 m. It shows, that the deflectors positioning is constraint by the mechanical design of the available deflector shield, because optimum positions above the software limit become overwritten. This effect can be seen for measured wind conditions with yaw angles of 20 ° to 30° (compare values in rows calculated and target). It also can be seen that the measured position of the deflector is within the positioning window of ± 3 cm from the calculated target position (compare values in rows target and system). The actual position that was measured by hand deviates from the systems measured position by about 2 cm (compares values in rows system and measured).

Yaw angle β	0 °	5 °	10 °	15 °	20 °	25 °	30 °
Calculated	4.5	4.6	4.7	4.75	4.8	4.8	4.8
System	4.59	4.61	4.70	4.78	4.78	4.78	4.78
Target	4.59	4.62	4.72	4.76	4.78	4.78	4.78
Measured	4.59	4.60	4.71	4.81	4.80	4.80	4.80

Table 6-2: Test results for 4.6 m for changing wind angles (in m)

A similar test for the reference trailer reference heights of 4.6 m and 4.2 m has been carried out.

Table 6-2 shows the results for a trailer reference height of 4.6 m, which was achieved by lowering the trailer top edge by 20 cm and leaving the cab roof height offset at 400 cm. That way the reference height position gets shifted to the lower end of the positioning range of the deflector and allows the testing of the adjustability higher than reference height. It shows, that the deflectors positioning is constraint by the mechanical design of the available deflector shield, because optimum positions below the software limit become overwritten (compare values in rows calculated and target). This effect can be seen for measured wind condition with yaw angle of 0 °. It also can be seen that the measured position of the deflector is within the positioning window of ±3 cm from the calculated target position (compare values in rows target and system). The actual position that has been measured by hand deviates from the systems measured position by up to 3 cm (compares values in rows system and measured).

Yaw angle β	0 °	5 °	10 °	15 °	20 °	25 °	30 °
Calculated	4.4	4.4	4.4	4.4	4.4	4.4	4.4
System	4.38	4.38	4.38	4.38	4.38	4.38	4.38
Target	4.38	4.38	4.38	4.38	4.38	4.38	4.38
Measured	4.41	4.41	4.41	4.41	4.41	4.41	4.41

Table 6-3: Test results for 4.2 m for changing wind angles (in m)

Table 6-3 shows the results for the trailer reference height of 4.2 m. Reducing the trailer reference height has been achieved by leaving the trailer height in the lowered position and by editing the cab roof height offset by minus 40 cm to 360 cm. The CFD simulation results for a trailer reference height of 4.2 m unfolded, that the optimum position does not change at all with changing wind conditions. Therefore the system remains static. The measured position of the deflector follows exactly the target position with no deviation. Comparing the deflector position measured by the system deflector position measured by hand shows deviations of 3cm.

The test results show that the concept of the system with all its software components explained in chapter 5 works but needs further improvements that could not be done within the given time span and with the given resources.

The developed control system allows to position the deflector within a positioning window of ± 3 cm. Depending on the sensor outputs the system decides if the deflector position needs to be adjusted. The difference of measured position in the system and the manually measured positioning of the deflector is likely to be originated in the combination of a convex deflector surface and the non-linear distance sensor curve used for the positioning measurement. The implementation of the evaluation algorithm for the five hole probe is robust in functionality and precisely calculates wind information. The accuracy for any wind vector component is higher than 97% (chapter 3-7).

Fluctuations in the pressure sensor outputs and the evaluation based on differentials of pressure sensor outputs have an impact on the systems usability. That issue was approached and handled with a ± 3 cm positioning corridor in the deflector adjustment function. Though that workaround stops the deflector shield from oscillating it leads to a reduction in positioning precision, which must not be a final solution. The precision of the positioning of the deflector shield needs to be improved in future work. The fluctuation in the sensor outputs also needs to be addressed, either with software workarounds such as averaging of multiple pressure readings of the same source or implementation of different sensor models.

Chapter 7

Conclusion

This chapter concludes the work that has been done, states the results that have been achieved and summarizes the limitations to be improved. It also provides recommendations for further work for development of a product rather than an engineering sample.

7. Conclusion

Historically HCV have been developed and manufactured to ensure a high payload capacity to fulfil the needs of the transport sector. The maximization of goods per vehicle ratio lead to designs of vehicles which neglected the aerodynamic streamlining and the effect of resulting drag forces on vehicles' fuel efficiency. The discussion of research results in chapter 1 has brought up that the longer the journey of a truck the higher is the percentage of fuel consumption that is directly related to aerodynamic drag. Over the last decade the greenhouse effect and fuel efficiency of vehicles has become a major focus in automotive development. The focus on fuel efficiency has led to different research projects in this area to improve aerodynamic streamlining of articulated trucks. These projects focussed their work on aerodynamic streamlining by changing geometries and/or additional aerodynamic enhancement components. Though many concepts of aerodynamic enhancements components have been developed most of them have been not taken up enthusiastically by the transport companies. This is because the changes on truck aerodynamics have to respect the needs of the transport companies such as low maintenance or quick return on investment to become accepted technology.

The development and effectiveness of drag reducing devices such as side skirts, cab roof deflectors and gap reduction devices is discussed in chapter 2. Studies and experiments on cab roof deflectors have raised awareness, that the optimum positioning of cab roof deflectors can reduce fuel consumption of articulated trucks by about 3-5 % overall.

7.1 Review and attainment of objectives

The realized project aimed to improve the aerodynamics of articulated trucks by reducing the shape based C_D coefficient. The goal was to develop a control system for automatically adjusting the cab roof deflectors. The control system should read the height of the trailer and the wind condition during the trucks journey to continuously adjust the deflector to the aerodynamically optimum position. A sensor array, a control circuit and a smart algorithm have been developed.

The sensor array features two infrared distance sensors and a five hole probe for measurement of the wind condition. The five hole probe has been calibrated by using the non-nulling calibration method of A.R Paul [25]. The obtained calibration parameters have been validated afterwards. The proven good calibration parameters have been implemented to calculate wind condition information as background for the automated CRD system.

The adjustment of the deflector shield position has been realized with a 24 V DC linear actuator and a control system which is based on standard automotive electronic components. The control system features an Atmel AT90CAN128 microcontroller, an automotive h-bridge motor driver and pressure sensors with analogue outputs to sense the pressures on the five hole probe holes. The control system processes the sensor output signals and makes decisions for the actuator outputs based on the acquired information.

The algorithm developed for the system is implemented in the microcontroller of the control system. The algorithm merges the sensor array including the five hole probe sensor with the actuator control outputs. The control logic consists of processing of sensor data, control of the actuator and a mathematical model for the positioning of the CRD which has been developed from CFD simulation results.

These three components create an automated CRD system that can adjust the deflector shield autonomously to changing wind conditions in real time. Its modular design in hardware and software allows easy adaption for different sensor types and deflector shield designs. Its processing and decision making logic can be easily modified for more precise calculation of the optimum deflector position, that could not be optimized because of a limited time frame for this project.

7.2 Original contribution of the work

The introduction and literature review gives an overview on the history of aerodynamic research and explains the latest research and engineering products. Most research projects on automotive aerodynamic drag issues focus on development of new ideas for aerodynamic streamlining. Though these projects concluded in new technologies for airflow optimisation many of the projects never make their way into the automotive industry. This is because the design, the availability, maintenance, return on investment rate and many other factors have to be accepted by the industry in general. If the design of the research result is not accepted the positive effect of the device might be neglected. This project did not aim to develop a new concept of airflow optimisation, but focussed on using available technology to improve available cab roof deflectors by automation.

Five hole probes are well known flow probes but were never used in automotive applications. This project has shown that five hole probes can be employed with automotive application to measure three dimensional wind information. Availability of three dimensional wind information can be of versatile use: It can be used for the positioning of aerodynamic devices

as shown in this project. It also can be used in further automotive systems that can be improved with real time wind information for example driving assistance systems.

Another original contribution can be found with the development of an algorithm for an automated CRD that is independent of the combination of cab roof height and trailer roof height. Latest automated cab roof deflector products available feature real time wind condition sensing, but they are limited with only certain combinations of truck and trailers. The algorithm for the calculation of the optimum deflector positioning that was developed during this project supports various combinations of different trucks and different trailers. The flexibility of this algorithm can be improved by employing more CFD datasets for compiling the optimum deflector positioning equations.

7.3 Major conclusions of the work

This project has shown how large numbers of CFD simulation results on aerodynamics of automotive vehicles can be compiled to be suitable for a microcontroller system with very limited computing power. This has been achieved by arranging the CFD results by the variables yaw angle and flow velocity and applying vector algebra to each optimum position of the CRD that were found with the CFD simulations. The vector algebra equations that describe surfaces that represent the optimum deflector position give good results but only need very little computing power.

This project has also shown that good results in enhancements of automotive aerodynamics can be achieved with limited resources. This should encourage researchers, engineers and scientists to perform projects that focus on further optimization of available and accepted aerodynamic enhancement components by automation.

7.4 Recommendation for future work

The primary goal of this project has been to develop a control system for automatic adjustment of articulated truck's cab roof deflectors. The research objectives for the development of the control system were fulfilled, but the system has some limitations that could not be addressed within the project timeframe. The system itself works as it was designed, but the electronic components are only suitable for a prototype in laboratory conditions. Before the system can be tested in real application the following issues need to be addressed and improved:

- The available deflector is sufficient for a working prototype. Its make is applicable for automotive application with a defined cab height and defined trailer height. If the idea

of a universally applicable automated CRD is taken forward it is recommended to redesign the deflector for a compatibility with a wider range of truck-trailer combinations.

- The CFD simulations that have been run allowed the development of a functional concept for CRD optimum position calculation that is applicable for microcontrollers with limited computing power. If the idea of an universally applicable automated CRD is taken forward the equations for the optimum deflector position calculation have to be recompiled. To do so also the set of CFD simulations with changed deflector geometry has to be run again.
- The aerodynamic model that has been developed allows to precisely calculate the optimum deflector position. This project has shown that the stream/vehicle velocity has no impact on the optimum deflector position for the CRD setup used. For further work it is recommended to expand the number of simulations for velocities lower than 40 mph and higher than 56 mph to get more precise optimum positions for crosswind situations.
- The sensors and the PCB for this project have been developed for laboratory conditions. The system performance test was successful and has shown the expected results. Taking the project forward requires changes in both sensor array and control system to sustain harsh environmental conditions such as rain and changing temperature conditions. Therefore all employed sensors have to be reviewed and are likely to be replaced in this process.
- A validity check of the initial CFD results that were used in this project brought up, that the CFD results might be faulty. These faulty CFD results might affect the performance of the system, since these simulation results do not match with the results of a real world application. The developed algorithm for data processing and the functionality of the developed control system is not affected by faulty simulation results. Though it is recommended to rerun the CFD simulations once the fault has been located and fixed. To update the system with the improved data the improved CFD results have to be processed into vector equations as described in chapter 5.1.3.4. The new parameters (Tables 5-2 to 5-4) have to be changed in the controller software. The modular approach of a core software and an independent config file makes the system easy to adapt for

changing geometries or changing CFD results. Changing the equations for the calculation of the optimum deflector position (Tables 5-2 to 5-4) in the config file will update the software to the new CFD results. The representation of the parameter tables in the config file can be found in Appendix A4 p. 155.

Chapter 6 has shown that the developed system is operational and performs well as an operational prototype. During the development of the system and the process of testing and debugging some issues and ideas for improvement came up that would require a complete redesign of the control system and its software or could not be made in time.

- The calculation of the wind vector information is proven good earlier but it contains two types of error. One error is based on the precision of the calibration parameters and is known from chapter 3.7. The other error results from the acquisition of the pressure sensor outputs. The pressure sensor outputs are read sequentially instead of simultaneous. This is because the pressure sensor outputs are interfaced directly to the ADC of the microcontroller. This issue can be improved by employing latching the sensor output signals before reading the signals into the controller. Alternatively a microcontroller could be used that has a latching component built in.
- The timer interval for triggering the sensor output acquisition has been set to a default value representing 500 ms (Chapter. 5.1.1). It is recommended to modulate the timer value with respect to the recurrence interval and the execution time of the wind calculation algorithm. The execution time can be found by implementing an output toggle command at the end of case 6 in the “ADC conversion completed” interrupt (Appendix A4 p. 126). The recurrence interval of the controller software can be calculated from the generated frequency on the toggle output. The timer interval for the sensor output acquisition can be adjusted accordingly.
- If the timer interval is adjusted to the recurrence interval of the controller software controller sleep modes can be used to reduce power consumption of the controller in operation. To do so the controller sleep mode [32, p. 46-50] has to be configured within the configuration section of the controller software (Chapter 5.1.1) and is activated at in the endless loop main routine. Exact timer interval triggering and sleep modes enable an exact timing of the controller software according to the application.

- If the measured recurrence interval of the controller software is found to slow for the application (for example with desired higher precision in the deflector positioning and therefore required higher measuring interval) it might be necessary to change the microcontroller. The used AT90CAN128 8-Bit controller does not feature a floating point calculation unit (FPU). The lack of a FPU requires the software compiler to compile floating point calculations to be executed with the logical unit of the microcontroller. This workaround in the compiler leads to a significantly increased calculation time for floating point calculations compared to calculation time with a controller that features a FPU. Applicable controllers that feature a FPU are likely to be ARM chip architecture controllers. Changing the controller architecture to ARM with FPU requires a full revision of the controller software and controller board but offers significant improvement in floating point calculation performance.

8. References

- [1] Department for Transport (2013). 'Transport Statistics Great Britain 2013' [online]
Available from:
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/264679/tsgb-2013.pdf
[Accessed last on 3. March 2015]
- [2] International Transport Forum. 'Permissible Maximum Dimensions of Trucks in Europe' [online]
Available from
<http://www.internationaltransportforum.org/IntOrg/road/pdf/dimensions.pdf>
[Accessed last on 3. March 2015]
- [3] Aurell, J., Wadman, T. (2007). 'Vehicle combinations based on the modular concept Background and analysis' [online]
Available from:
http://www.modularsystem.eu/download/facts_and_figures/20080522att02.pdf
[Accessed last on 27. April 2015]
- [4] Mohamed-Kassim, Z., Filippone, A. (2010) 'Fuel savings on a heavy vehicle via aerodynamic drag reduction' Transportation Research Part D, P. 275- 284
- [5] McCallen, R., Browand, F., Hammache, M., Leonard, A., Brady, M., Salari, K., Rutlege, W., Ross, J., Storms, B., Heineck, J.T. (1999) 'Progress in Reducing Aerodynamic Drag for Higher Efficiency of Heavy Duty Trucks (Class 7-8)' from Society of Automotive Engineers Government/ Industry Meeting Washington D.C. April 26-28, 1999
- [6] Malviya, V. M. (2011) 'Effects of a Novel Aerodynamic Intervention for Heavy Commercial Vehicles on Fuel Saving and Stability'. University of Huddersfield, Huddersfield, UK.
- [7] Drollinger, R. A. (1987) 'Heavy Duty Truck Aerodynamics' SAE/SP-87/688, Society of Automotive Engineers, Warrendale

- [8] Garry, K. P. (1985) 'A review of commercial vehicle aerodynamic drag reduction techniques' Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 199:215
- [9] McCallen, R., Flowers, D., Dunn, T., Owens, J. Browand, F., Hammache, M., Leonard, A., Brady, M., Salari, K., Rutledge, W., Ross, J., Storms, B., Heineck, J.T., Driver, D., Bell, J., Walker, S., Zilliac, G. (2000) 'Aerodynamic Drag of Heavy Vehicles (Class 7-8): Simulation and Benchmarking' (2000-01-2209) from Government/Industry Meeting Washington D.C. June 19-21, 2000
- [10] Road speed limiters (RSL's)
Available from
<http://www.transportsfriend.org/road/rsl.html>
[Accessed last on 27. April 2015]
- [11] Dominy, R. G., (1997) 'Fuel saving by the optimized use of cab roof fairings for typical articulated vehicle operating cycles' Proc Instn Mech Engrs Voll 211 Part D, P. 219 -227
- [12] Modi, V.J., Hill, S. St., Yokomizo, T. (1995) 'Drag reduction of trucks through boundary-layer control' Journal of Wind Engineering and Industrial Aerodynamics 54/55 P. 583-594
- [13] Wood, R. M., Bauer, S. X. S. (2003) 'Simple and Low-Cost Aerodynamic Drag Reduction Devices for Tractor-Trailer Trucks' (2003-01-3377) SAE International
- [14] Cooper, K. R., Leuschen, J. (2005) 'Model and Full-Scale Wind Tunnel Tests of Second-Generation Aerodynamic Fuel Saving Devices for Tractor-Trailers' (05CV-64) SAE International
- [15] Department for Transport (2006). 'Smoothing the Flow at TNT Express and Somerfield using Truck Aerodynamic Styling'. *Freight Best Practice* [online]
Available from:
http://ukerc.rl.ac.uk/pdf/Truck_Aerodynamic_Styling.pdf
[Accessed last 3. March 2015]

- [16] Chowdhury, H., Moria, H., Ali, A., Khan, I., Alam, F., Watkins, S. (2013) 'A study on aerodynamic drag of a semi-trailer truck' *Procedia Engineering* 56, P. 201-205.
- [17] Blake, N. M., Garry, K. P., Getley, B. 'Air deflector arrangement', UK Patent GB2465393, Date of Filing: 17. Nov 2008
- [18] Hatcher Components LTD. 'New Active Freddie Tests Show Fuel Cost Savings' [online]
Available from:
<http://www.hatchercomp.co.uk/wp-content/uploads/2012/09/001-Tesco-Report-Flyer-2pp.pdf>
[Accessed last 3. March 2015]
- [19] Rogers, M. C. C., Yee, J., Kashiwagi, B., Guilfoyle, M., Brzustiwicz, J., McFarlane, R., Banceu, B. 'Dynamically adjustable aerodynamic vehicle devices', US Patent US2011/0241377, Published 6. Oct 2011
- [20] Emborg, U., Hammar, J., Kristiansson, G., 'An air guiding device and a method of reducing the air resistance of a ground vehicle', International Patent WO2013/117538A1, Published 15 Aug 2013
- [21] Lipták, B. 'Aviation Safety, Variable Control Valve Failure'. CONTROL's August 2009 edition. F [online]
Available from:
<http://www.controlglobal.com/articles/2009/safetyfailure0908/>
[Accessed last 3. March 2015]
- [22] Argüelles Diaz, K. M., Fernández Oro, J. M., Blanco Marigorta, E. (2009) 'Cylindrical three-hole pressure probe calibration for large angular range' *Flow Measurement and Instrumentation* 20 (2009), P. 57-68
- [23] Bryer, D. W., Pankhurst R. C., (1971) 'Pressure-probe methods for determining wind speed and flow direction' Her Majesty's Stationary Office, London
- [24] Krause, L. N. and Dudzindki, T. J., Flow-direction measurement with fixed position probes in subsonic flows over a range of Reynolds number, NASA TMX-52576, 1969.

- [25] Akshoy Ranjan Paul, Ravi Ranjan Upadhyay, Anuj Jain (2011) ‘A novel calibration algorithm for five-hole pressure probe’ International Journal of Engineering, Science and Technology, Vol. 3, No.2, 2011, pp. 89-95
- [26] Freescale Semiconductor. ‘MPXV7002 Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated’ [online]
Available from:
http://www.freescale.com/files/sensors/doc/data_sheet/MPXV7002.pdf
- [27] Microsoft Corporation. ‘Microsoft Excel Online Help – LINEST function’ [online]
Available from:
<https://support.office.com/en-gb/article/LINEST-function-806c2ef0-8347-472d-b915-fd60c880022a>
[Accessed last on 27. Apr 2015]
- [28] Microsoft Corporation. ‘Microsoft Excel Online Help – RGP function’ [online]
Available from:
<https://support.office.com/de-ch/article/RGP-Funktion-84d7d0d9-6e50-4101-977a-fa7abf772b6d>
[Accessed last on 27. Apr 2015]
- [29] Industrial Devices (GB) LTD. ‘IDM 8A Series’ Datasheet [online]
Available from:
<http://www.actuators-electric.co.uk/wp-content/uploads/idm8-series-acme.pdf>
[Accessed last on 27. Apr 2015]
- [30] Sharp Electronics ‘Sharp GP2Y0A02YK0F’ Datasheet [online]
Available from:
https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf
[Accessed last on 27. Apr 2015]
- [31] ST Microelectronics. ‘ST VNH3SP30-E’ Datasheet [online]
Available from:
<https://www.pololu.com/file/0J51/vnh3sp30.pdf>
[Accessed last on 27. Apr 2015]

- [32] Atmel Corporation. ‘Atmel 8-bit AVR Microcontroller with 32K/64K/128K Bytes of ISP flash and CAN Controller’ [online]
Available from:
<http://www.atmel.com/images/doc7679.pdf>
[Accessed last on 27. Apr 2015]
- [33] Traco Electronic AG. ‘TracoPower DC/DC Converters TSR-1 Series, 1 A’ [online]
Available from:
<http://www.adafruit.com/datasheets/tsr1.pdf>
[Accessed last on 27. Apr 2015]
- [34] Atmel Corporation ‘Atmel AVR Studio 4.19-730’ [online]
Software download available from:
www.atmel.com/Images/AvrStudio4Setup.exe
[Accessed last on 27. Apr 2015]
- [35] WinAVR library [online]
Software download available from:
<http://sourceforge.net/projects/winavr/files/WinAVR/20100110/>
[Accessed last on 27. Apr 2015]
- [36] WinAVR library documentation [online]
available from:
<http://www.nongnu.org/avr-libc/user-manual/index.html>
[Accessed last on 27. Apr 2015]
- [37] Laser & Co. solutions GmbH. ‘mySmartUSB light’ technical description [online]
Available from:
http://www.myavr.info/download/produkte/mysmartusb_light/techb_mySmartUSB-light_de_en.pdf
[Accessed last on 27. Apr 2015]

Appendices

A1 Five hole probe calibration parameters

Sector1															
Pitch	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0,1965	227,7747	-1,2780	-39,2661	-7,8622	0,5384	-507,4178	11,6292	62,4756	-23,2594	-0,6431	380,7628	1,6198	-140,0242	-7,4680
se	0,2340	158,6646	1,4852	38,3185	8,7651	0,7751	359,2329	9,3534	57,2168	25,9187	2,5171	290,2089	4,0467	95,7795	10,2108
r ²	0,8844	3,1474	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	23,4958	43,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	3258,5216	425,9611	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector1															
Yaw	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	0,2337	-785,0286	0,6970	126,1190	55,3549	-2,3112	1596,3407	-49,9698	-166,9966	64,9674	7,0635	-1205,8648	8,9267	383,1680	-34,7429
se	0,8091	548,6432	5,1355	132,4834	30,3088	2,6801	1242,1845	32,3431	197,8489	89,6238	8,7038	1003,5078	13,9931	331,1942	35,3077
r ²	0,7113	10,8833	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	7,5682	43,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	12549,9109	5093,1925	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector1															
CP static	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0,0030	7,9097	-0,0221	-1,4913	-0,6264	0,0172	-15,8834	0,6638	2,0496	-0,6369	-0,0866	12,1113	0,0454	-6,1882	1,8436
se	0,0069	4,6943	0,0439	1,1336	0,2593	0,0229	10,6284	0,2767	1,6928	0,7668	0,0745	8,5862	0,1197	2,8338	0,3021
r ²	0,9730	0,0931	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	110,7051	43,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	13,4395	0,3729	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector1															
CP_total	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	0,0101	2,2998	0,0335	-0,3240	-0,4755	-0,0111	-4,1015	0,3077	0,8134	-0,5966	-0,1548	1,5033	0,0887	0,7382	-0,3977
se	0,0033	2,2619	0,0212	0,5462	0,1250	0,0110	5,1211	0,1333	0,8157	0,3695	0,0359	4,1371	0,0577	1,3654	0,1456
r ²	0,9702	0,0449	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	100,0462	43,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	2,8197	0,0866	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV

Figure A 1: Five hole probe calibration parameters for Sector 1

Sector2															
Pitch	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	26,9184	0,4634	-12,0259	14,5837	1,5559	-48,6160	-4,6777	11,7757	-9,6557	34,6515	2,3388	-0,9292	4,1847	-0,1672	
Se	66,3493	0,4832	48,0554	2,2769	9,4602	174,4824	1,7434	83,1614	11,6814	44,2898	164,1325	62,9754	7,4892	8,2763	
r ²	0,9534	5,1942	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	112,5529	77,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	42513,8274	2077,4769	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector2															
Yaw	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-69,9127	0,9802	-21,0897	3,3510	-5,4480	189,0450	-2,7119	32,8824	-13,8694	-5,1924	-145,9494	-0,5018	68,8812	3,8515	12,3142
Se	32,6619	0,2379	23,6563	1,1209	4,6570	85,8928	0,8582	40,9380	21,8026	5,7504	80,7978	1,8353	31,0010	3,6867	4,0742
r ²	0,9301	2,5570	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	73,1354	77,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	6694,3881	503,4380	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector2															
CP_static	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	4,2727	0,0033	0,1114	-0,0070	-0,0795	-12,7290	-0,0187	-0,1219	0,3182	-0,0344	13,4723	-0,0821	-7,9608	0,0503	2,2888
Se	1,3006	0,0095	0,9420	0,0446	0,1854	3,4203	0,0342	1,6302	0,2290	0,8682	3,2174	0,0731	1,2345	0,1468	0,1622
r ²	0,9631	0,1018	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	143,3856	77,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	20,8113	0,7983	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector2															
CP_total	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	1,2982	0,0176	-0,9085	0,0660	0,3262	-2,7005	-0,0457	1,0253	-0,2337	-0,5041	0,7005	-0,0266	0,9630	0,0205	-0,4271
Se	0,8429	0,0061	0,6105	0,0289	0,1202	2,2166	0,0221	1,0565	0,1484	0,5627	2,0851	0,0474	0,8000	0,0951	0,1051
r ²	0,9212	0,0660	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	64,3365	77,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	3,9220	0,3353	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV

Figure A 2: Five hole probe calibration parameters for Sector 2

Sector3															
Pitch	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0,3026	-149,9085	-3,8576	56,1612	-4,9910	-2,3119	-311,7698	-7,2460	91,4571	51,9179	-2,9218	-214,3579	10,5955	-78,0819	14,8876
Se	0,4172	114,8562	3,0226	54,9060	8,2135	1,9586	265,1851	10,9059	92,7009	50,1440	3,2381	208,4137	8,6978	64,0825	6,1113
r ²	0,9441	1,7414	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	20,5008	17,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	870,3246	51,5504	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector3															
Yaw	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0,7562	125,7029	2,3714	-146,6531	15,8551	-0,7826	365,2274	21,5165	-261,6972	-156,3267	10,0458	351,8351	-13,3665	129,4798	9,3471
Se	0,7454	205,2268	5,4007	98,1069	14,6760	3,4997	473,8370	19,4869	165,6395	89,5981	5,7859	372,3969	15,5413	114,5037	10,9198
r ²	0,9700	3,1115	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	39,2197	17,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	5315,8832	164,5855	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector3															
static	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0,0290	7,9927	-0,1234	0,2409	-0,1815	-0,1038	18,2650	-0,1659	0,1717	0,1892	0,0987	16,4131	0,1830	8,4418	2,3002
Se	0,0292	8,0439	0,2117	3,8453	0,5752	0,1372	18,5720	0,7638	6,4922	3,5118	0,2268	14,5961	0,6091	4,4880	0,4280
r ²	0,9635	0,1220	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	32,0218	17,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	6,6677	0,2528	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector3															
total	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0,0069	-10,2338	-0,0638	2,6885	-0,6089	-0,0305	-24,3292	-0,7103	4,4046	2,1813	-0,2634	-20,3953	0,2467	-6,9112	-0,8690
Se	0,0282	7,7528	0,2040	3,7062	0,5544	0,1322	17,9001	0,7362	6,2574	3,3847	0,2186	14,0680	0,5871	4,3256	0,4125
r ²	0,7422	0,1175	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	3,4961	17,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	0,6763	0,2349	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV

Figure A 3: Five hole probe calibration parameters for Sector 3

Sector4															
Pitch	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-164,9583	-1,2536	42,9833	-7,4823	-28,7215	-396,8196	-0,5949	61,6365	-38,0366	64,7605	-299,7738	-6,7172	-74,5534	0,8049	-6,4497
Se	230,6513	0,5474	58,8314	2,6102	14,6771	503,4628	1,7898	104,9845	18,0725	56,4660	384,2483	5,6681	119,1607	8,7525	12,5837
r ²	0,9688	4,2526	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	126,2562	57,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	31666,3443	1021,1557	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector4															
Yaw	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	61,6208	-0,8125	-6,0930	-0,8519	-12,5945	52,9905	-0,7876	-14,6832	-20,9460	-5,0776	-30,3073	-1,7426	6,8369	2,2395	-17,3790
Se	134,1605	0,3184	34,2198	1,5183	8,5371	292,8439	1,0410	61,0652	10,5121	32,8440	223,5017	3,2969	69,3110	5,0910	7,3194
r ²	0,9120	2,4619	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	42,2161	57,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	3582,2924	345,4854	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector4															
static	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-7,8853	0,0126	2,3790	-0,0316	-0,9271	-19,4173	-0,0305	4,2519	-1,1733	2,3481	-14,7654	-0,3018	-1,7409	0,3515	1,0470
Se	6,5118	0,0155	1,6610	0,0737	0,4144	14,2140	0,0505	2,9640	0,5102	1,5942	10,8482	0,1600	3,3642	0,2471	0,3553
r ²	0,9320	0,1195	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	55,7987	57,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	11,1549	0,8139	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector4															
total	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-3,5651	0,0097	1,0871	0,0383	0,1771	-6,4020	0,0320	2,0429	0,3039	1,0172	-4,5453	-0,0255	-1,8167	0,0525	-0,4001
Se	4,6758	0,0111	1,1926	0,0529	0,2975	10,2063	0,0363	2,1283	0,3664	1,1447	7,7895	0,1149	2,4156	0,1774	0,2551
r ²	0,7851	0,0858	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	14,8714	57,0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	1,5328	0,4197	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV

Figure A 4: Five hole probe calibration parameters for Sector 4

Sector5															
Pitch	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0.4371	-1.4920	0.1998	-0.0141	-1.6895	1.6814	0.4604	0.2196	0.8611	-0.8078	4.0673	6.9392	-3.3788	-11.2975	-7.3199
Se	1.4900	1.7484	1.5490	2.3244	1.5336	1.6473	1.4144	1.1287	1.7413	4.2382	3.6816	4.5600	2.5329	2.8904	2.2992
r ²	0.8416	5.7490	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	12.5267	33.0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	5796.2917	1090.6874	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector5															
Yaw	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	0.7851	0.4842	-0.4451	-0.3343	0.0393	0.0510	-0.0129	0.0467	0.0696	0.4104	-2.0807	-1.7473	9.5566	0.7818	1.9431
Se	0.4256	0.4994	0.4425	0.6640	0.4381	0.4705	0.4040	0.3224	0.4974	1.2106	1.0517	1.3026	0.7235	0.8256	0.6568
r ²	0.9786	1.6422	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	107.5603	33.0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	4061.0046	88.9954	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector5															
CP_static	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	-0.0568	-0.0970	-0.0302	0.0248	-0.1007	0.0657	0.0819	0.0703	0.0601	0.0117	0.1840	0.3105	-0.1338	-0.2388	0.7636
Se	0.0719	0.0844	0.0747	0.1122	0.0740	0.0795	0.0682	0.0545	0.0840	0.2045	0.1776	0.2200	0.1222	0.1395	0.1109
r ²	0.2369	0.2774	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	0.7317	33.0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	0.7881	2.5391	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
Sector5															
CP_total	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
coeff	0.0193	-0.0254	0.0219	0.0159	-0.0580	0.0301	0.0098	-0.0046	0.0014	-0.0550	-0.0515	0.0340	-0.0909	-0.0257	-0.0621
Se	0.0205	0.0240	0.0213	0.0319	0.0211	0.0226	0.0194	0.0155	0.0239	0.0582	0.0506	0.0626	0.0348	0.0397	0.0316
r ²	5.6806	33.0000	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV
	0.4960	0.2058	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV	#NV

Figure A 5: Five hole probe calibration parameters for Sector 5

A2 Control system main circuit board

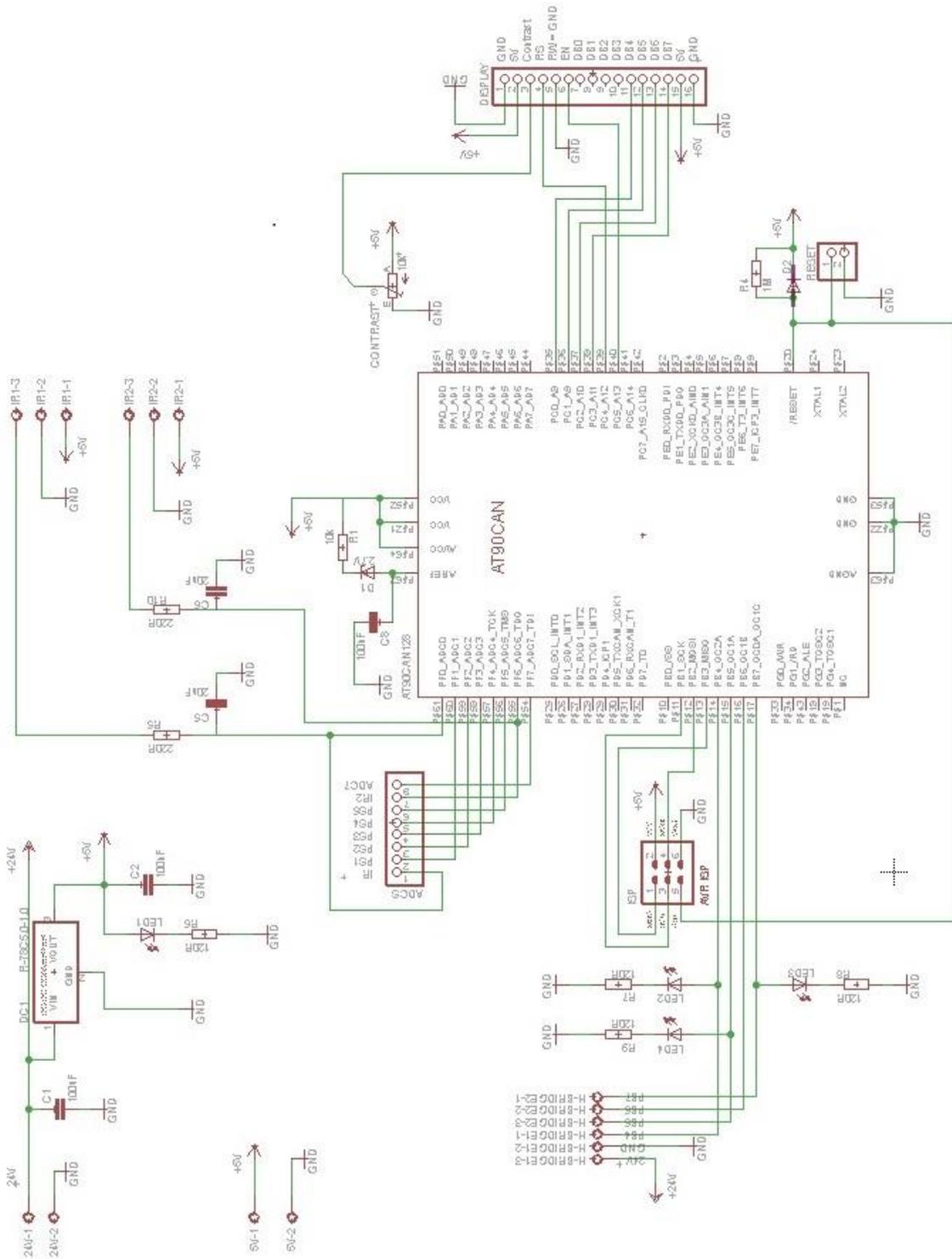


Figure A 6: Control system main circuit board schematic

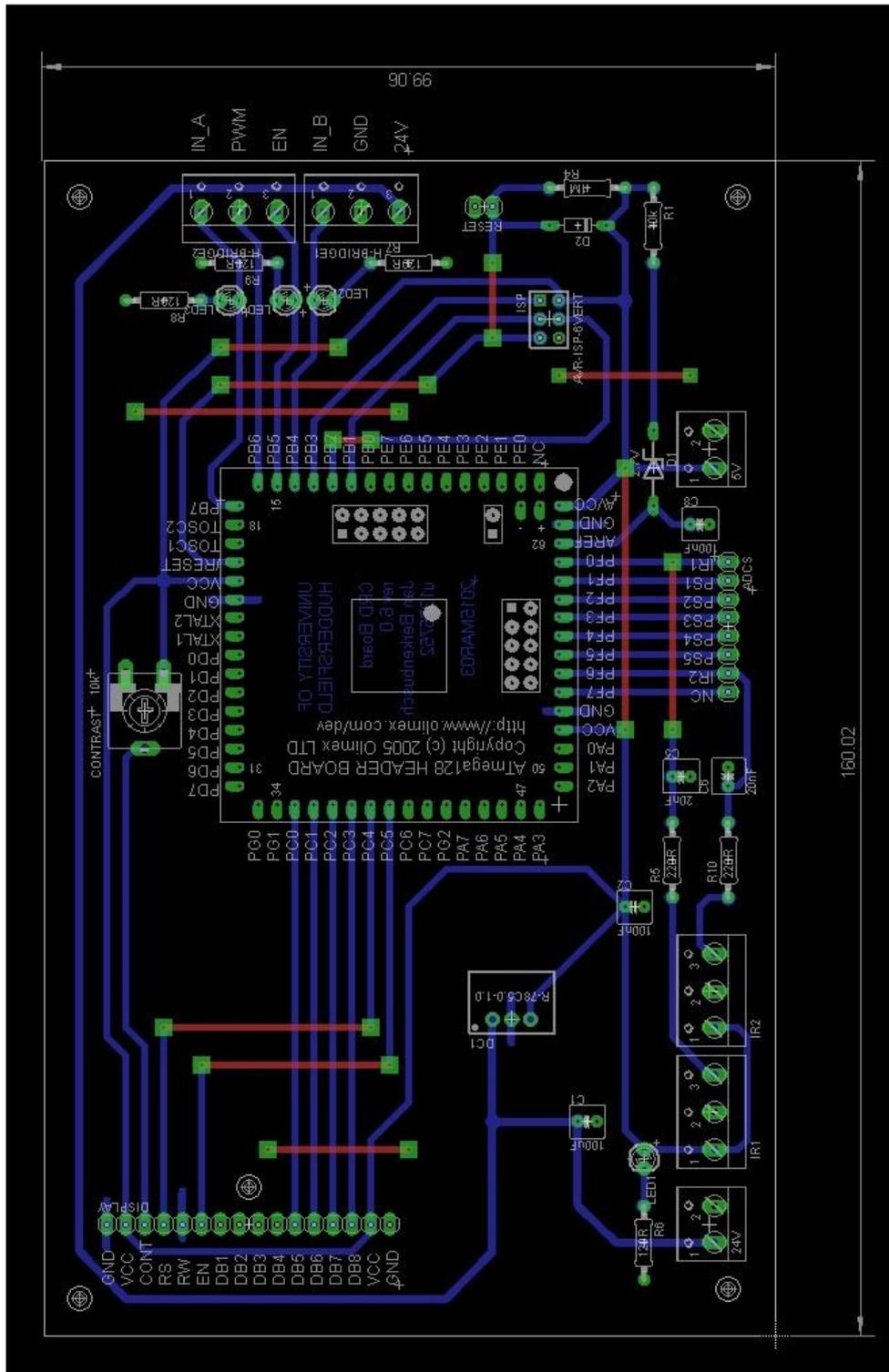


Figure A 7: Control system main circuit board layout

A3 Table with compiled CFD Results

Model	Yaw	Speed	fairing position	Drag force	Drag coef	side force	Side coef	
4.2m	0	40	0.2 m above trailer edge	1933,33	0,96			
			0.1 m above trailer edge	1944,89	0,97			
			Parallel to trailer edge	1947,86	0,97			
			0.1 m below trailer edge	1953,11	0,97			
		0.2 m below trailer edge	1966,78	0,98				
		50	0.2 m above trailer edge	3010,24	0,96			
			0.1 m above trailer edge	3021,52	0,96			
			Parallel to trailer edge	3016,91	0,96			
	0.1 m below trailer edge		3029,89	0,96				
	56	0.2 m below trailer edge	3038,3	0,97				
		0.2 m above trailer edge	3740,65	0,95				
		0.1 m above trailer edge	3755,44	0,95				
		Parallel to trailer edge	3777,71	0,96				
	4,2	10	40	0.2 m above trailer edge	3561,64	1,769	2270,25	1,127
				0.1 m above trailer edge	3603,93	1,79	2250,41	1,11
				Parallel to trailer edge	3682,76	1,829	2151,44	1,068
0.1 m below trailer edge				3734,91	1,855	2142	1,06	
0.2 m below trailer edge			3728,89	1,852	2062	1,024		
50			0.2 m above trailer edge	5512,02	1,752	3566,95	1,133	
			0.1 m above trailer edge	5586,88	1,776	3564,99	1,133	
			Parallel to trailer edge	5697,88	1,81	3372,35	1,07	
			0.1 m below trailer edge	5776,93	1,836	3172,72	1,01	
0.2 m below trailer edge			5764,5	1,832	3245,86	1,03		
56			0.2 m above trailer edge	6890,77	1,746	4460,6	1,13	
			0.1 m above trailer edge	6986,1	1,77	4433,1	1,12	
		Parallel to trailer edge	7126,23	1,8	4296,84	1,1		
		0.1 m below trailer edge	7223,17	1,83	4040,28	1,02		
0.2 m below trailer edge		7228,13	1,831	4128,69	1,04			
20		40	0.2 m above trailer edge	5407,61	2,68	4554,5	2,26	
			0.1 m above trailer edge	5542,34	2,75	4577,11	2,27	
			Parallel to trailer edge	5687,05	2,82	4569,75	2,26	
			0.1 m below trailer edge	5654,1	2,8	4444,66	2,2	
		0.2 m below trailer edge	5638,01	2,8	4417,33	2,19		
		50	0.2 m above trailer edge	8415,96	2,67	7146,32	2,27	
			0.1 m above trailer edge	8626,44	2,74	7220,58	2,29	
			Parallel to trailer edge	8842,28	2,81	7153,44	2,27	
			0.1 m below trailer edge	8783,82	2,79	6974,33	2,21	
	0.2 m below trailer edge	8757,28	2,78	7002,19	2,22			
	56	0.2 m above trailer edge	10578,01	2,68	9072,99	2,3		
		0.1 m above trailer edge	10774,1	2,73	9022,1	2,28		
Parallel to trailer edge		11055,17	2,8	8935,43	2,26			
0.1 m below trailer edge		11001,19	2,78	8739,45	2,21			
0.2 m below trailer edge	10934,99	2,77	8768,74	2,22				
30	40	0.2 m above trailer edge	7478,07	3,71	6112,98	3,03		
		0.1 m above trailer edge	7573,19	3,76	6249,11	3,1		
		Parallel to trailer edge	8264,26	4,1	7074,97	3,51		
		0.1 m below trailer edge	7883,21	3,91	6467,34	3,21		
	0.2 m below trailer edge	7901,43	3,92	6539,78	3,24			
	50	0.2 m above trailer edge	11577,97	3,68	9409,1	2,99		
		0.1 m above trailer edge	11655,76	3,7	9502,4	3,02		
		Parallel to trailer edge	12784,22	4,06	10999,64	3,49		
		0.1 m below trailer edge	12332,1	3,92	10179,76	3,23		
	0.2 m below trailer edge	12336,33	3,92	10272,58	3,26			
	56	0.2 m above trailer edge	14487,24	3,67	11803,09	2,99		
		0.1 m above trailer edge	14665,44	3,71	12008,25	3,04		
Parallel to trailer edge		15996,17	4,05	13893,06	3,52			
0.1 m below trailer edge		15280,24	3,87	12610,94	3,19			
0.2 m below trailer edge	15404,2	3,9	12872,99	3,26				

Figure A 8: CFD results for a trailer reference height of 4.2 m

Model	Yaw	Speed	fairing position	Drag force	Drag coef	side force	Side coef
4.6m	0	40	0.2 m above trailer edge	2018,56	0,91		
			0.1 m above trailer edge	1966,43	0,89		
			Parallel to trailer edge	1949,37	0,88		
		0.1 m below trailer edge	1945,03	0,88			
		0.2 m below trailer edge	1974,54	0,89			
		0.2 m above trailer edge	3108,85	0,90			
	50	40	0.1 m above trailer edge	3074,42	0,89		
			Parallel to trailer edge	3018,59	0,87		
			0.1 m below trailer edge	3017,52	0,87		
		0.2 m below trailer edge	3065,75	0,89			
		56	0.2 m above trailer edge	3889,80	0,90		
			0.1 m above trailer edge	3839,11	0,88		
Parallel to trailer edge	3772,27		0,87				
10	50	40	0.1 m below trailer edge	3786,89	1,70	2601,16	1,17
			0.2 m below trailer edge	3897,94	1,75	2437,06	1,09
			0.2 m above trailer edge	5797,82	1,67	4560,14	1,31
		56	0.1 m above trailer edge	5694,28	1,64	4545,58	1,31
			Parallel to trailer edge	5732,08	1,65	4408,51	1,27
			0.1 m below trailer edge	5870,50	1,69	4098,38	1,18
	50	40	0.2 m below trailer edge	6063,82	1,75	3812,67	1,1
			0.2 m above trailer edge	7261,36	1,67	5561,17	1,27
			0.1 m above trailer edge	7133,64	1,64	5683,81	1,3
		56	Parallel to trailer edge	7165,63	1,64	5522,04	1,27
			0.1 m below trailer edge	7323,72	1,68	5081,2	1,16
			0.2 m below trailer edge	7572,59	1,74	4781,9	1,1
20	50	40	0.2 m above trailer edge	5798,86	2,61	5314,36	2,39
			0.1 m above trailer edge	5826,36	2,62	5356,45	2,41
			Parallel to trailer edge	5916,63	2,66	5340,87	2,4
		56	0.1 m below trailer edge	5961,76	2,68	5280,15	2,38
			0.2 m below trailer edge	6019,14	2,71	5145,15	2,32
			0.2 m above trailer edge	9014,89	2,60	8312,24	2,39
	50	40	0.1 m above trailer edge	9061,54	2,61	8300,23	2,39
			Parallel to trailer edge	9147,97	2,64	8363,3	2,41
			0.1 m below trailer edge	9190,41	2,65	8180,69	2,36
		56	0.2 m below trailer edge	9559,54	2,75	8170,33	2,35
			0.2 m above trailer edge	11259,23	2,59	10502,03	2,41
			0.1 m above trailer edge	11294,48	2,59	10419,32	2,39
30	50	40	Parallel to trailer edge	11448,73	2,63	10464,93	2,4
			0.1 m below trailer edge	11489,11	2,64	10311,3	2,37
			0.2 m below trailer edge	11997,61	2,76	10257,72	2,36
		56	0.2 m above trailer edge	8150,98	3,67	6889,37	3,1
			0.1 m above trailer edge	8233,28	3,71	6955,65	3,13
			Parallel to trailer edge	8280,31	3,73	7066,18	3,18
	50	40	0.1 m below trailer edge	8417,96	3,79	7231,28	3,26
			0.2 m below trailer edge	8256,44	3,72	6966,09	3,14
			0.2 m above trailer edge	12749,52	3,68	10854,53	3,13
		56	0.1 m above trailer edge	12898,54	3,72	10998,77	3,17
			Parallel to trailer edge	12903,36	3,72	11150,42	3,21
			0.1 m below trailer edge	13106,88	3,78	11343,26	3,27
50	40	0.2 m below trailer edge	12836,87	3,70	10850,99	3,13	
		0.2 m above trailer edge	15937,80	3,66	13719,3	3,15	
		0.1 m above trailer edge	16094,51	3,70	13830,04	3,18	
	56	Parallel to trailer edge	16147,26	3,71	13978,07	3,21	
		0.1 m below trailer edge	16363,97	3,76	14159,41	3,25	
		0.2 m below trailer edge	16087,16	3,70	13619,06	3,13	

Figure A 9: CFD results for a trailer reference height of 4.6 m

Model	Yaw	Speed	fairing position	Drag force	Drag coef	side force	Side coef		
4.8m	0	40	0.2 m above trailer edge	2070,83	0,89				
			0.1 m above trailer edge	2050,85	0,88				
			Parallel to trailer edge	1986,79	0,86				
			0.1 m below trailer edge	1977,2	0,85				
					0.2 m below trailer edge	1954,5	0,84		
		50	0.2 m above trailer edge	3220,81	0,89				
			0.1 m above trailer edge	3178,96	0,88				
			Parallel to trailer edge	3136,29	0,87				
			0.1 m below trailer edge	3044,29	0,84				
					0.2 m below trailer edge	3008,58	0,83		
		56	0.2 m above trailer edge	3220,81	0,89				
			0.1 m above trailer edge	3178,96	0,88				
Parallel to trailer edge	3136,29		0,87						
0.1 m below trailer edge	3044,29		0,84						
			0.2 m below trailer edge	3008,58	0,83				
4,8	10	40	0.2 m above trailer edge	3832,77	1,65	3168,05	1,36		
			0.1 m above trailer edge	3807,41	1,64	3143,62	1,35		
			Parallel to trailer edge	3776,91	1,62	3246,69	1,4		
			0.1 m below trailer edge	3766,57	1,62	3100,62	1,33		
					0.2 m below trailer edge	3846,62	1,65	2941,7	1,26
		50	0.2 m above trailer edge	5936,76	1,63	4933,76	1,36		
			0.1 m above trailer edge	5911,85	1,63	4796,02	1,32		
			Parallel to trailer edge	5849,74	1,61	4924,11	1,36		
			0.1 m below trailer edge	5847,22	1,61	4870,48	1,34		
					0.2 m below trailer edge	5935,42	1,63	4580,79	1,26
		56	0.2 m above trailer edge	7428,45	1,63	6117,87	1,34		
			0.1 m above trailer edge	7389,93	1,62	6119,73	1,34		
	Parallel to trailer edge		7309,63	1,61	6301,34	1,38			
	0.1 m below trailer edge		7309,05	1,6	6033,13	1,32			
				0.2 m below trailer edge	7450,86	1,63	5777,69	1,27	
	20	40	0.2 m above trailer edge	6032	2,6	5781,45	2,49		
			0.1 m above trailer edge	6012,98	2,59	5764,12	2,48		
			Parallel to trailer edge	6111,5	2,63	5813,42	2,5		
			0.1 m below trailer edge	6085,89	2,62	5702,97	2,45		
					0.2 m below trailer edge	6165,03	2,65	5622,68	2,42
		50	0.2 m above trailer edge	9383,72	2,59	8912,79	2,45		
			0.1 m above trailer edge	9373,01	2,58	9066,51	2,5		
			Parallel to trailer edge	9495,53	2,62	9038,84	2,49		
			0.1 m below trailer edge	9511,69	2,62	8983,46	2,47		
					0.2 m below trailer edge	9604,4	2,65	8863,05	2,44
		56	0.2 m above trailer edge	11826,68	2,6	11412,64	2,51		
			0.1 m above trailer edge	11708,28	2,57	11428,83	2,51		
	Parallel to trailer edge		11854,19	2,6	11346,8	2,49			
	0.1 m below trailer edge		11834,3	2,6	11203,33	2,46			
				0.2 m below trailer edge	11932,38	2,62	11037,9	2,42	
	30	40	0.2 m above trailer edge	8733,55	3,76	7677,14	3,31		
			0.1 m above trailer edge	8620,54	3,71	7516,46	3,24		
			Parallel to trailer edge	8673,83	3,74	7546,8	3,25		
			0.1 m below trailer edge	8660,43	3,73	7536,8	3,25		
					0.2 m below trailer edge	8819,17	3,8	7736,05	3,33
		50	0.2 m above trailer edge	13595,66	3,75	11989,53	3,31		
0.1 m above trailer edge			13445,89	3,71	11714	3,23			
Parallel to trailer edge			13459,77	3,71	11755,45	3,24			
0.1 m below trailer edge			13488,18	3,72	11773,88	3,24			
				0.2 m below trailer edge	13713,95	3,78	12031,76	3,32	
56		0.2 m above trailer edge	17010,29	3,74	15017,39	3,3			
		0.1 m above trailer edge	16840,78	3,7	14754,32	3,24			
	Parallel to trailer edge	16885,33	3,71	14734,22	3,24				
	0.1 m below trailer edge	16884,89	3,71	14762,44	3,24				
			0.2 m below trailer edge	17192,7	3,78	15102,18	3,32		

Figure A 10: CFD results for a trailer reference height of 4.8 m


```

struct DATA
{
    uint16_t      PS1;
    uint16_t      PS2;
    uint16_t      PS3;
    uint16_t      PS4;
    uint16_t      PS5;
    uint16_t      IR;
    uint16_t      IR2;
    uint16_t      SORTIME [5];

    double P1;
    double P2;
    double P3;
    double P4;
    double P5;

    double CPalpha;
    double CPbeta;

    double CPstatic;
    double CPtotal;

    double Pstatic;
    double Ptotal;

    double Alpha;
    double Beta;
    double V;

    double CRD_REFERENCE;
    double CRD_TARGET;
    double CRD_HEIGHT;
    double CRD_DELTA_H;
};

struct DATA VALUES;

// ===== CREATING GLOBAL VARIABLES FOR USE IN ISR =====
volatile uint16_t      pitot_read = 0;
volatile uint8_t       channel = 0;
volatile uint8_t       flag = 0;
volatile char          *buffer [15];
volatile double        temp_double;
volatile int16_t       temp_int16;
volatile uint8_t       sortme_counter = 0;
volatile uint16_t      avg_PS1 [6];
volatile uint16_t      avg_PS2 [6];
volatile uint16_t      avg_PS3 [6];
volatile uint16_t      avg_PS4 [6];
volatile uint16_t      avg_PS5 [6];
volatile uint16_t      avg_IR2 [6];
volatile uint8_t       avgctr = 0;
volatile int8_t        Offset_PS1;
volatile int8_t        Offset_PS2;
volatile int8_t        Offset_PS3;
volatile int8_t        Offset_PS4;
volatile int8_t        Offset_PS5;
volatile struct DATA  VALUES;

volatile uint8_t cycle=0;

// FUNCTION PROTOTYPES:
void DEFLECTOR_UP (void);
void DEFLECTOR_DOWN (void);
void DEFLECTOR_STOP(void);
void lcd_send_float (double);
void sortme (void);
void wind_calculation(void);
void optimum_position(void);
void ps_referencing(void);

```

```

// ===== Main Programm=====
int main (void)
{

// ===== CREATING LOCAL VARIABLES FOR FUNCTION MAIN. NOT TO BE USED IN ISR =====
volatile double temp_double;
volatile int16_t temp_int16;

//PIN CONFIGURATION
//DDRA
//DDRB |= (1<<HB_IN_A | 1<<HB_IN_B | 1<<HB_EN | 1<<HB_PWM);           //H-Bridge Outputs
//DDRC
//DDRD
//DDRE
//DDRF

// ===== INITIALIZING 16BIT TIMER TO 500 ms INTERRUPT =====
// for register descriptions see datasheet p. 136 - 144
TCCR1A = 0x00; // 0000 0000
TCCR1B = (1<<WGM12 | 1<<CS12 ); // 0000 1100 CTC Mode, Prescaler 256
TCCR1C = 0x00; // 0000 0000
TIMSK1 = (1<<OCIE1A); // 0000 0010 compare Match RegA enabled
// => interrupt generated
OCR1A = 0x3D09; //3D09 = 500ms, 7A13 = 1000ms, F4A4 = 2000ms
//OCR1AH = 0x00; // 8MHz : 256 = 15625 = 500ms
//OCR1AL = 0xFF; //F424 or 7A12 or 3D09
//CS11+CS10 = PS 64, => 6250 = 50ms

// ===== INITIALIZING ADC CONVERTER =====
// for register descriptions see datasheet p. 289 ff.

ADCSRA = (1<<ADEN | 1<<ADPS2 | 1<<ADPS1 | 1<<ADIE); // (1<<ADEN | 1<<ADPS2 | 1<<ADPS0)
ADCSRA &= ~(1<<ADATE);
ADCSRA |= (1<<ADSC); // 1001 1111 Write 1 to Bit 6 to start ADC conversion,
// ADC-Prescaler=64
ADMUX |= (1<<MUX1); // 0000 0010 ADC Multiplexer. last 4 bits are channel selection

//LCD_Display intialization
lcd_init();
//show installed version for 1s
lcd_string("CRD_v6.0a");
sei();
_delay_ms (1000);
lcd_setcursor(0,2);

ADCSRA |= (1<<ADSC);

```

```

// FINDING TRAILER REFERENCE HEIGHT
while (VALUES.IR >= 100)
{
    DEFLECTOR_UP();           //moving top edge over the trailer top edge
    lcd_setcursor(0,1);
    lcd_string ("UP ");
    lcd_send_float (VALUES.IR);
    lcd_setcursor(0,2);
    lcd_send_float (VALUES.CRD_HEIGHT);
    ADCSRA |= (1<<ADSC);
} //end while

DEFLECTOR_STOP();
_delay_ms(2000);
lcd_clear();

while (VALUES.IR < 150)
{
    DEFLECTOR_DOWN();        // move down till trailer edge is detected
    lcd_setcursor(0,1);
    lcd_string ("DOWN ");
    lcd_send_float (VALUES.IR);
    lcd_setcursor(0,2);
    lcd_send_float (VALUES.CRD_HEIGHT);
    ADCSRA |= (1<<ADSC);
} //end while

DEFLECTOR_STOP();

VALUES.CRD_REFERENCE= VALUES.CRD_HEIGHT-10;
lcd_clear();
lcd_string ("Ref. Height:");
lcd_setcursor(0,2);
lcd_send_float (VALUES.CRD_REFERENCE);
_delay_ms (2000);

//auto referencing ADC Inputs for pressure sensors to 900 ADC tics
Offset_PS1 = Ref_Volts_PS1 - VALUES.PS1 +1;
Offset_PS2 = Ref_Volts_PS2 - VALUES.PS2 +1;
Offset_PS3 = Ref_Volts_PS3 - VALUES.PS3 +1;
Offset_PS4 = Ref_Volts_PS4 - VALUES.PS4 +1;
Offset_PS5 = Ref_Volts_PS5 - VALUES.PS5 +1;

while (1)
{
    if (flag == 1)
    {
        flag = 0;
        wind_calculation();
    } //end if flag = 1
} //end while

} //end main

// CREATE INTERRUPT ROUTINE FOR STARTING AN ADC CONVERSION

// ===== CREATE INTERRUPT ROUTINE FOR 1s TIMER =====
ISR (TIMER1_COMPA_vect) //TIMER1_OVF_vect TIMER1_COMPA_vect
{
    ADCSRA |= (1<<ADSC);           //START ADC CONVERSION
} // end Timer ISR

```

```

ISR (ADC_vect)
{
    cycle++;

    switch (channel)
    {
        case 0: //IR1
            {
                temp_int16 = ADCL;
                temp_int16 |= (ADCH<<8);
                VALUES.SORTME[avgctr] = temp_int16;
                sortme();
                channel++;
                ADMUX++;
                ADCSRA |= (1<<ADSC);
                break;
            }

        case 1: //PS1
            {
                temp_int16 = ADCL;
                temp_int16 |= (ADCH<<8);

                avg_PS1[avgctr] = (temp_int16+Offset_PS1) ;
                VALUES.PS1 = (avg_PS1[0] +avg_PS1[1] +avg_PS1[2] +avg_PS1[3]
                    +avg_PS1[4])/5;

                channel++;
                ADMUX++;
                ADCSRA |= (1<<ADSC);
                break;
            }

        case 2: //PS2
            {
                temp_int16 = ADCL;
                temp_int16 |= (ADCH<<8);
                avg_PS2[avgctr] = (temp_int16+Offset_PS2) ;
                VALUES.PS2 = (avg_PS2[0] + avg_PS2[1] + avg_PS2[2] + avg_PS2[3] +
                    avg_PS2[4])/5;

                channel++;
                ADMUX++;
                ADCSRA |= (1<<ADSC);
                break;
            }

        case 3: //PS3
            {
                temp_int16 = ADCL;
                temp_int16 |= (ADCH<<8);
                avg_PS3[avgctr] = (temp_int16+Offset_PS3) ;
                VALUES.PS3 = (avg_PS3[0] + avg_PS3[1] + avg_PS3[2] + avg_PS3[3] +
                    avg_PS3[4])/5;

                channel++;
                ADMUX++;
                ADCSRA |= (1<<ADSC);
                break;
            }

        case 4: //PS4
            {
                temp_int16 = ADCL;
                temp_int16 |= (ADCH<<8);
                avg_PS4[avgctr] = (temp_int16+Offset_PS4) ;
                VALUES.PS4 = (avg_PS4[0] + avg_PS4[1] + avg_PS4[2] + avg_PS4[3] +
                    avg_PS4[4])/5;

                channel++;
                ADMUX++;
                ADCSRA |= (1<<ADSC);
                break;
            }
    }
}

```

```

case 5: //PS5
{
temp_int16 = ADCL;
temp_int16 |= (ADCH<<8);
avg_PS5[avgctr] = (temp_int16+Offset_PS5) ;
VALUES.PS5 = (avg_PS5[0] + avg_PS5[1] + avg_PS5[2] + avg_PS5[3] +
avg_PS5[4])/5;

channel++;
ADMUX++;
ADCSRA |= (1<<ADSC);
break;
}

case 6: //IR2
{
temp_int16 = ADCL;
temp_int16 |= (ADCH<<8);
avg_IR2[avgctr] = (temp_int16) ;
channel=0;
ADMUX &=0xF0;

VALUES.IR2 = ((avg_IR2[0] + avg_IR2[1] + avg_IR2[2] + avg_IR2[3] +
avg_IR2[4])/5);
temp_double = (VALUES.IR2 * 0.00283 * 27.27);
VALUES.CRD_HEIGHT = 119.0 - temp_double + cab_roof_height;
flag = 1;
avgctr++;
break;
}

} //end switch

if (avgctr >= 5)
avgctr = 0;
cycle++;
if (cycle >200)
cycle = 0;
} //end isr

void wind_calculation(void)
{
volatile uint8_t sector = 1;
volatile uint16_t highest = 0;

cli();
// DEFLECTOR_STOP();
//finding the sector for the given set of readings
highest = VALUES.PS1;

if (highest < VALUES.P2)
{
highest = VALUES.P2;
sector = 2;
}
if (highest < VALUES.P3)
{
highest = VALUES.P3;
sector = 3;
}
if (highest < VALUES.P4)
{
highest = VALUES.P4;
sector = 4;
}
if (highest < VALUES.P5)
{
highest = VALUES.P5;
sector = 5;
}
}

```

```

// transforming ADC readings into absolute Pressures.
//VALUES.P1
temp_int16   =   (VALUES.PS1 - Ref_Volts_PS1);
temp_double  =   temp_int16*283;
temp_double  =   temp_double/10000;
VALUES.P1    =   1013.0 + temp_double;
//VALUES.P2
temp_int16   =   (VALUES.PS2 - Ref_Volts_PS2);
temp_double  =   temp_int16*283;
temp_double  =   temp_double/10000;
VALUES.P2    =   1013.0 + temp_double;
//VALUES.P3
temp_int16   =   (VALUES.PS3 - Ref_Volts_PS3);
temp_double  =   temp_int16*283;
temp_double  =   temp_double/10000;
VALUES.P3    =   1013.0 + temp_double;
//VALUES.P4
temp_int16   =   (VALUES.PS4 - Ref_Volts_PS4);
temp_double  =   temp_int16*283;
temp_double  =   temp_double/10000;
VALUES.P4    =   1013.0 + temp_double;
//VALUES.P5
temp_int16   =   (VALUES.PS5 - Ref_Volts_PS5);
temp_double  =   temp_int16*283;
temp_double  =   temp_double/10000;
VALUES.P5    =   1013.0 + temp_double;

/* === Sending values to LCD for debugging === */
lcd_setcursor (0,1);
lcd_send_float (VALUES.PS1);
lcd_string (" ");
lcd_send_float (VALUES.PS2);
lcd_string (" ");
lcd_send_float (VALUES.PS3);
lcd_setcursor (0,2);
lcd_send_float (VALUES.PS4);
lcd_string (" ");
lcd_send_float (VALUES.PS5);
lcd_string (" ");
lcd_send_float (cycle);

```

```

//calculation of pressure coefficients based on related sectors:
switch (sector)
{
case 1:
{
temp_double = ((VALUES.P4 + VALUES.P5 + VALUES.P2)/3);
temp_double = ( VALUES.P1 - temp_double);
VALUES.CPalpha = ((VALUES.P1 - VALUES.P5) / temp_double);
//Cp_alpha = (P1-P5) / (P1 -((P4 + P5 + P2)/3))

VALUES.CPbeta = ((VALUES.P2 - VALUES.P4) / temp_double);
//Cp_beta = (P2-P4) / (P1 -((P4 + P5 + P2)/3))

//Calculation for angle alpha:
temp_double = PS1_pitch_a0;

temp_double = temp_double + (PS1_pitch_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS1_pitch_a2 * VALUES.CPbeta) ;

temp_double = temp_double + (PS1_pitch_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS1_pitch_a4 * VALUES.CPbeta *
VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a5 * VALUES.CPalpha *
VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS1_pitch_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_pitch_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.Alpha = temp_double + (PS1_pitch_a14 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

//Calculation for angle beta:
temp_double = PS1_yaw_a0 ;

temp_double = temp_double + (PS1_yaw_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS1_yaw_a2 * VALUES.CPbeta) ;

temp_double = temp_double + (PS1_yaw_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS1_yaw_a4 * VALUES.CPbeta *
VALUES.CPbeta );

temp_double = temp_double + (PS1_yaw_a5 * VALUES.CPalpha *
VALUES.CPbeta );

```

```

temp_double = temp_double + (PS1_yaw_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS1_yaw_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_yaw_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS1_yaw_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_yaw_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_yaw_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS1_yaw_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_yaw_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.Beta = temp_double + (PS1_yaw_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_total:
```

```

temp_double = PS1_total_a0 ;

temp_double = temp_double + (PS1_total_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS1_total_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS1_total_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS1_total_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS1_total_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS1_total_a6 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta );

temp_double = temp_double + (PS1_total_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS1_total_a8 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha);

temp_double = temp_double + (PS1_total_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS1_total_a10 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_total_a11 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS1_total_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_total_a13 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPtotal =temp_double+ (PS1_total_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```

//Calculation for P_static:
temp_double = PS1_static_a0 ;

temp_double = temp_double + (PS1_static_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS1_static_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS1_static_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS1_static_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS1_static_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS1_static_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS1_static_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS1_static_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS1_static_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS1_static_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_static_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS1_static_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS1_static_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPstatic=temp_double+ (PS1_static_a14 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

//Calculation for Windspeed V:
VALUES.Pstatic = (VALUES.CPstatic * (-1) * (VALUES.P1 - ((VALUES.P2 + VALUES.P4
+ VALUES.P5)/3))) + ((VALUES.P2 + VALUES.P4 + VALUES.P5)/3);

VALUES.Ptotal = (VALUES.CPtotal * (-1) * (VALUES.P1 - ((VALUES.P2 + VALUES.P4
+ VALUES.P5)/3))) + VALUES.P1;

temp_double = (VALUES.Ptotal - VALUES.Pstatic)*2/rho_air;

VALUES.V = sqrt (temp_double);
break;
}

```

```

case 2:
{
temp_double = ((VALUES.P1 + VALUES.P5 + VALUES.P3)/3);
temp_double = ( VALUES.P2 - temp_double);
VALUES.CPalpha = ((VALUES.P1 - VALUES.P3) / temp_double);
//Cp_alpha = (P1 - P3 ) / (P2 -(( P1 + P5 + P3 )/3))

VALUES.CPbeta = ((VALUES.P2 - VALUES.P5) / temp_double);
//Cp_beta = (P2 - P5) / (P2 -(( P1 + P5 + P3 )/3))

//Calculation for angle alpha:
temp_double = PS2_pitch_a0 ;

temp_double = temp_double + (PS2_pitch_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS2_pitch_a2 * VALUES.CPbeta) ;

temp_double = temp_double + (PS2_pitch_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS2_pitch_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS2_pitch_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS2_pitch_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS2_pitch_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS2_pitch_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS2_pitch_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS2_pitch_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_pitch_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS2_pitch_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_pitch_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.Alpha = temp_double + (PS2_pitch_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

//Calculation for angle beta:
temp_double = PS2_yaw_a0 ;

temp_double = temp_double + (PS2_yaw_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS2_yaw_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS2_yaw_a4 * VALUES.CPbeta *
VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a5 * VALUES.CPalpha *
VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

```

```

temp_double = temp_double + (PS2_yaw_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_yaw_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

VALUES.Beta = temp_double + (PS2_yaw_a14 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_total:
```

```

temp_double = PS2_total_a0 ;

temp_double = temp_double + (PS2_total_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS2_total_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS2_total_a4 * VALUES.CPbeta *
VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a5 * VALUES.CPalpha *
VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS2_total_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

VALUES.CPtotal =temp_double+ (PS2_total_a14 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_static:
```

```
temp_double = PS2_static_a0 ;
```

```

temp_double = temp_double + (PS2_static_a1 * VALUES.CPalpha);
temp_double = temp_double + (PS2_static_a2 * VALUES.CPbeta );
temp_double = temp_double + (PS2_static_a3 * VALUES.CPalpha *
VALUES.CPalpha);
temp_double = temp_double + (PS2_static_a4 * VALUES.CPbeta * VALUES.CPbeta
);
temp_double = temp_double + (PS2_static_a5 * VALUES.CPalpha * VALUES.CPbeta
);
temp_double = temp_double + (PS2_static_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS2_static_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS2_static_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);
temp_double = temp_double + (PS2_static_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS2_static_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS2_static_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS2_static_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS2_static_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);
VALUES.CPstatic=temp_double+ (PS2_static_a14 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

```

```

//Calculation for Windspeed V:

```

```

VALUES.Pstatic = (VALUES.CPstatic * (-1) * (VALUES.P2 - ((VALUES.P1 + VALUES.P3
+ VALUES.P5)/3))) + ((VALUES.P1 + VALUES.P3 + VALUES.P5)/3);
VALUES.Ptotal = (VALUES.CPtotal * (-1) * (VALUES.P2 - ((VALUES.P1 + VALUES.P3
+ VALUES.P5)/3))) + VALUES.P2;
temp_double = (VALUES.Ptotal - VALUES.Pstatic)*2/rho_air;
VALUES.V = sqrt (temp_double);
break;
}

```

```

case 3:
{
temp_double = ((VALUES.P4 + VALUES.P5 + VALUES.P2)/3);
VALUES.CPalpha = (VALUES.P5 - VALUES.P3) / (VALUES.P3-temp_double);
//Cp_alpha = (P5 - P3) / (P3 - ((P4 +
P5 + P2 )/3))
//temp_double = ((VALUES.PS4+VALUES.PS5+VALUES.PS2)/3);
VALUES.CPbeta = (VALUES.P2-VALUES.P4) / (VALUES.P3-temp_double);
//Cp_beta = (P2 - P4) / (P3 - ((P4 +
P5 + P2 )/3))

//Calculation for angle alpha:
temp_double = PS3_pitch_a0 ;
temp_double = temp_double + (PS3_pitch_a1 * VALUES.CPalpha);
temp_double = temp_double + (PS3_pitch_a2 * VALUES.CPbeta) ;
temp_double = temp_double + (PS3_pitch_a3 * VALUES.CPalpha *
VALUES.CPalpha);
temp_double = temp_double + (PS3_pitch_a4 * VALUES.CPbeta * VALUES.CPbeta
);
temp_double = temp_double + (PS3_pitch_a5 * VALUES.CPalpha * VALUES.CPbeta
);
temp_double = temp_double + (PS3_pitch_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS3_pitch_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );
temp_double = temp_double + (PS3_pitch_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);
temp_double = temp_double + (PS3_pitch_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );
temp_double = temp_double + (PS3_pitch_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS3_pitch_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS3_pitch_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS3_pitch_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);
VALUES.Alpha = temp_double + (PS3_pitch_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

//Calculation for angle beta:
temp_double = PS3_yaw_a0 ;
temp_double = temp_double + (PS3_yaw_a1 * VALUES.CPalpha);
temp_double = temp_double + (PS3_yaw_a2 * VALUES.CPbeta) ;
temp_double = temp_double + (PS3_yaw_a3 * VALUES.CPalpha *
VALUES.CPalpha);
temp_double = temp_double + (PS3_yaw_a4 * VALUES.CPbeta *
VALUES.CPbeta );
temp_double = temp_double + (PS3_yaw_a5 * VALUES.CPalpha *
VALUES.CPbeta );

```

```

temp_double = temp_double + (PS3_yaw_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS3_yaw_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_yaw_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS3_yaw_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_yaw_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_yaw_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS3_yaw_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_yaw_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.Beta = temp_double + (PS3_yaw_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_total:
```

```

temp_double = PS3_total_a0 ;

temp_double = temp_double + (PS3_total_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS3_total_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS3_total_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS3_total_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS3_total_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS3_total_a6 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta );

temp_double = temp_double + (PS3_total_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS3_total_a8 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha);

temp_double = temp_double + (PS3_total_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS3_total_a10 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_total_a11 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS3_total_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_total_a13 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPtotal =temp_double+ (PS3_total_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_static:
```

```

temp_double = PS3_static_a0 ;

temp_double = temp_double + (PS3_static_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS3_static_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS3_static_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS3_static_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS3_static_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS3_static_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS3_static_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_static_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS3_static_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_static_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_static_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS3_static_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS3_static_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPstatic=temp_double+ (PS3_static_a14 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

```

```

//Calculation for Windspeed V:

```

```

VALUES.Pstatic = (VALUES.CPstatic * (-1) * (VALUES.P3 - ((VALUES.P2 + VALUES.P4
+ VALUES.P5)/3))) + ((VALUES.P2 + VALUES.P4 + VALUES.P5)/3);

VALUES.Ptotal = (VALUES.CPtotal * (-1) * (VALUES.P3 - ((VALUES.P2 + VALUES.P4
+ VALUES.P5)/3))) + VALUES.P3;

temp_double = (VALUES.Ptotal - VALUES.Pstatic)*2/rho_air;
VALUES.V = sqrt (temp_double);
break;
}

```

```

case 4:
{
temp_double = ((VALUES.P1 + VALUES.P5 + VALUES.P3)/3);
VALUES.CPalpha = (VALUES.P1 - VALUES.P3) / (VALUES.P4-temp_double);
//Cp_alpha = (P1 - P3) / (P4 - ((P1 + P5 +
P3 )/3))
//temp_double = ((VALUES.PS1+VALUES.PS5+VALUES.PS3)/3);
VALUES.CPbeta = (VALUES.P5-VALUES.P4) / (VALUES.P4-temp_double);
//Cp_beta = (P5 - P4) / (P4 - ((P1 + P5
+ P3 )/3))

//Calculation for angle alpha:
temp_double = PS4_pitch_a0 ;

temp_double = temp_double + (PS4_pitch_a1 * VALUES.CPalpha);
temp_double = temp_double + (PS4_pitch_a2 * VALUES.CPbeta) ;
temp_double = temp_double + (PS4_pitch_a3 * VALUES.CPalpha *
VALUES.CPalpha);
temp_double = temp_double + (PS4_pitch_a4 * VALUES.CPbeta * VALUES.CPbeta
);
temp_double = temp_double + (PS4_pitch_a5 * VALUES.CPalpha * VALUES.CPbeta
);
temp_double = temp_double + (PS4_pitch_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS4_pitch_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );
temp_double = temp_double + (PS4_pitch_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);
temp_double = temp_double + (PS4_pitch_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );
temp_double = temp_double + (PS4_pitch_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS4_pitch_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS4_pitch_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS4_pitch_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);
VALUES.Alpha = temp_double + (PS4_pitch_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

//Calculation for angle beta:
temp_double = PS4_yaw_a0 ;

temp_double = temp_double + (PS4_yaw_a1 * VALUES.CPalpha);
temp_double = temp_double + (PS4_yaw_a2 * VALUES.CPbeta) ;
temp_double = temp_double + (PS4_yaw_a3 * VALUES.CPalpha *
VALUES.CPalpha);
temp_double = temp_double + (PS4_yaw_a4 * VALUES.CPbeta *
VALUES.CPbeta );
temp_double = temp_double + (PS4_yaw_a5 * VALUES.CPalpha *
VALUES.CPbeta );
temp_double = temp_double + (PS4_yaw_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS4_yaw_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

```

```

temp_double = temp_double + (PS4_yaw_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS4_yaw_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_yaw_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_yaw_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS4_yaw_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_yaw_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.Beta = temp_double + (PS4_yaw_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_total:
```

```

temp_double = PS4_total_a0 ;

temp_double = temp_double + (PS4_total_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS4_total_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS4_total_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS4_total_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS4_total_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS4_total_a6 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta );

temp_double = temp_double + (PS4_total_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS4_total_a8 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha);

temp_double = temp_double + (PS4_total_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS4_total_a10 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_total_a11 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS4_total_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_total_a13 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPtotal =temp_double+ (PS4_total_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_static:
```

```

temp_double = PS4_static_a0 ;

temp_double = temp_double + (PS4_static_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS4_static_a2 * VALUES.CPbeta );

```

```

temp_double = temp_double + (PS4_static_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS4_static_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS4_static_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS4_static_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS4_static_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_static_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS4_static_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_static_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_static_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS4_static_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS4_static_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPstatic=temp_double+ (PS4_static_a14 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

```

```

//Calculation for Windspeed V:

```

```

VALUES.Pstatic = (VALUES.CPstatic * (-1) * (VALUES.P1 - ((VALUES.P1 + VALUES.P3
+ VALUES.P5)/3))) + ((VALUES.P1 + VALUES.P3 + VALUES.P5)/3);

VALUES.Ptotal = (VALUES.CPtotal * (-1) * (VALUES.P1 - ((VALUES.P1 + VALUES.P3
+ VALUES.P5)/3))) + VALUES.P4;

temp_double = (VALUES.Ptotal - VALUES.Pstatic)*2/rho_air;
VALUES.V = sqrt (temp_double);
break;
}

```

```

case 5:
{
temp_double = ((VALUES.P1 + VALUES.P2 + VALUES.P3 + VALUES.P4)/4);
VALUES.CPalpha = (VALUES.P1 - VALUES.P3) / (VALUES.P5-temp_double);
//Cp_alpha = (P1 - P3) / (P5 - ((P1 + P2
+ P3 + P4)/4))
//temp_double = ((VALUES.PS1+VALUES.PS2+VALUES.PS3+VALUES.PS4)/4);
VALUES.CPbeta = (VALUES.P2-VALUES.P4) / (VALUES.P5-temp_double);
//Cp_beta = (P2 - P4) / (P5 - ((P1 + P2
+ P3 + P4)/4))

//Calculation for angle alpha:
temp_double = PS5_pitch_a0 ;
temp_double = temp_double + (PS5_pitch_a1 * VALUES.CPalpha);
temp_double = temp_double + (PS5_pitch_a2 * VALUES.CPbeta) ;
temp_double = temp_double + (PS5_pitch_a3 * VALUES.CPalpha *
VALUES.CPalpha);
temp_double = temp_double + (PS5_pitch_a4 * VALUES.CPbeta * VALUES.CPbeta
);
temp_double = temp_double + (PS5_pitch_a5 * VALUES.CPalpha * VALUES.CPbeta
);
temp_double = temp_double + (PS5_pitch_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS5_pitch_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );
temp_double = temp_double + (PS5_pitch_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);
temp_double = temp_double + (PS5_pitch_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );
temp_double = temp_double + (PS5_pitch_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS5_pitch_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );
temp_double = temp_double + (PS5_pitch_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );
temp_double = temp_double + (PS5_pitch_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);
VALUES.Alpha = temp_double + (PS5_pitch_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

//Calculation for angle beta:
temp_double = PS5_yaw_a0 ;
temp_double = temp_double + (PS5_yaw_a1 * VALUES.CPalpha);
temp_double = temp_double + (PS5_yaw_a2 * VALUES.CPbeta );
temp_double = temp_double + (PS5_yaw_a3 * VALUES.CPalpha *
VALUES.CPalpha);
temp_double = temp_double + (PS5_yaw_a4 * VALUES.CPbeta *
VALUES.CPbeta );
temp_double = temp_double + (PS5_yaw_a5 * VALUES.CPalpha *
VALUES.CPbeta );

```

```

temp_double = temp_double + (PS5_yaw_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS5_yaw_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_yaw_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS5_yaw_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_yaw_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_yaw_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS5_yaw_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_yaw_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.Beta = temp_double + (PS5_yaw_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_total:
```

```

temp_double = PS5_total_a0 ;

temp_double = temp_double + (PS5_total_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS5_total_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS5_total_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS5_total_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS5_total_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS5_total_a6 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta );

temp_double = temp_double + (PS5_total_a7 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS5_total_a8 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha);

temp_double = temp_double + (PS5_total_a9 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta );

temp_double = temp_double + (PS5_total_a10 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_total_a11 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS5_total_a12 * VALUES.CPalpha * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_total_a13 * VALUES.CPalpha * VALUES.CPalpha
* VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPtotal =temp_double+ (PS5_total_a14 * VALUES.CPbeta * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

```

```
//Calculation for P_static:
```

```

temp_double = PS5_static_a0 ;

temp_double = temp_double + (PS5_static_a1 * VALUES.CPalpha);

temp_double = temp_double + (PS5_static_a2 * VALUES.CPbeta );

temp_double = temp_double + (PS5_static_a3 * VALUES.CPalpha *
VALUES.CPalpha);

temp_double = temp_double + (PS5_static_a4 * VALUES.CPbeta * VALUES.CPbeta
);

temp_double = temp_double + (PS5_static_a5 * VALUES.CPalpha * VALUES.CPbeta
);

temp_double = temp_double + (PS5_static_a6 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS5_static_a7 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_static_a8 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha);

temp_double = temp_double + (PS5_static_a9 * VALUES.CPbeta *
VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_static_a10 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_static_a11 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPbeta );

temp_double = temp_double + (PS5_static_a12 * VALUES.CPalpha *
VALUES.CPbeta * VALUES.CPbeta * VALUES.CPbeta );

temp_double = temp_double + (PS5_static_a13 * VALUES.CPalpha *
VALUES.CPalpha * VALUES.CPalpha * VALUES.CPalpha);

VALUES.CPstatic=temp_double+ (PS5_static_a14 * VALUES.CPbeta * VALUES.CPbeta
* VALUES.CPbeta * VALUES.CPbeta );

//Calculation for Windspeed V:
VALUES.Pstatic = ((VALUES.CPstatic * (-1) * (VALUES.P5 - (( VALUES.P1 +
VALUES.P2 + VALUES.P3 + VALUES.P4)/4))) + ((VALUES.P1 + VALUES.P2 + VALUES.P4 +
VALUES.P5)/4));

VALUES.Ptotal = ((VALUES.CPtotal * (-1) * (VALUES.P5 - (( VALUES.P1 +
VALUES.P2 + VALUES.P3 + VALUES.P4)/4))) + VALUES.P5);

temp_double = ((VALUES.Ptotal) - (VALUES.Pstatic))*200/rho_air;
VALUES.V = sqrt (temp_double);
break;
}

default:
{
printf (buffer, "Calc. Error");
lcd_string (buffer);
_delay_ms(5000);
break;
}

} // end switch (sector)

sei();
optimum_position();

}

```

```

void optimum_position(void)
{
double MPH;
/*
=== MANUAL INPUT FOR TESTING EQUATION SELECTION ===
VALUES.CRD_REFERENCE = 4.72;
VALUES.Alpha = 13.0;
VALUES.Beta = 13.0;
VALUES.V = 20.0;
*/

if (VALUES.Beta < 0.0)
VALUES.Beta = VALUES.Beta * (-1);
//m/s in mph conversion: *3600/1000/1.6 = *2.25
MPH = (VALUES.V*2.25);

/*selecting height in steps. 4.2m - 4.6m - 4.8m
4.2 0 10 20 30 alpha
#-----#-----#-----#-----#
| 4.4 | 4.4 | 4.4 | 4.4 |
#----- 1 ----- 3 ----- 5 -----# > height in m
| 4.4 | 4.4 | 4.4 | 4.4 |
#----- 2 ----- 4 ----- 6 -----#
| 4.4 | 4.4 | 4.4 | 4.4 |
#-----#-----#-----#-----#
speed
4.6 0 10 20 30 alpha
#-----#-----#-----#-----#
| 4.5 | 4.7 | 4.8 | 4.8 |
#----- 1 ----- 3 ----- 5 -----# > height in m
| 4.6 | 4.7 | 4.8 | 4.8 |
#----- 2 ----- 4 ----- 6 -----#
| 4.5 | 4.7 | 4.8 | 4.8 |
#-----#-----#-----#-----#
speed
4.8 0 10 20 30 alpha
#-----#-----#-----#-----#
| 4.6 | 4.7 | 4.9 | 4.9 |
#----- 1 ----- 3 ----- 5 -----# > height in m
| 4.6 | 4.7 | 4.9 | 4.9 |
#----- 2 ----- 4 ----- 6 -----#
| 4.6 | 4.7 | 4.9 | 4.9 |
#-----#-----#-----#-----#
speed
*/

if ( (VALUES.CRD_REFERENCE > 410 ) & (VALUES.CRD_REFERENCE < 440)) //4.1 to 4.4 = ref 4.2
{
if (VALUES.Alpha < 10)
VALUES.CRD_TARGET = eq4_2__1;

if ((VALUES.Alpha >= 10) & (VALUES.Alpha < 20))
VALUES.CRD_TARGET = eq4_2__3;
}

```

```

        if ((VALUES.Alpha >= 20) & (VALUES.Alpha < 30))
            VALUES.CRD_TARGET = eq4_2__5;
    }

if ((VALUES.CRD_REFERENCE >= 440) & (VALUES.CRD_REFERENCE < 470)) // 4.4 to 4.7 = ref 4.6
{
    if (VALUES.Alpha < 10)
        VALUES.CRD_TARGET = eq4_6__1;

    if ((VALUES.Alpha >= 10) & (VALUES.Alpha < 20))
        VALUES.CRD_TARGET = eq4_6__3;

    if ((VALUES.Alpha >= 20) & (VALUES.Alpha < 30))
        VALUES.CRD_TARGET = eq4_6__5;
}

if (VALUES.CRD_REFERENCE >= 470 ) // greater 4.7 = ref 4.8
{
    if (VALUES.Alpha < 10)
        VALUES.CRD_TARGET = eq4_8__1;

    if ((VALUES.Alpha >= 10) & (VALUES.Alpha < 20))
        VALUES.CRD_TARGET = eq4_8__3;

    if ((VALUES.Alpha >= 20) & (VALUES.Alpha < 30))
        VALUES.CRD_TARGET = eq4_8__5;
}

VALUES.CRD_DELTA_H = VALUES.CRD_TARGET - VALUES.CRD_HEIGHT; //calculating height difference

//if (VALUES.CRD_DELTA_H < 0.0)
//    VALUES.CRD_DELTA_H = (VALUES.CRD_DELTA_H*(-1)); // height difference delta_h

if (VALUES.CRD_TARGET > IR_upper_limit)
    VALUES.CRD_TARGET = IR_upper_limit;
if (VALUES.CRD_TARGET < IR_lower_limit)
    VALUES.CRD_TARGET = IR_lower_limit;

//checks for delta_h. if delta_j is greater than 3cm readjustments take place
if ( (VALUES.CRD_DELTA_H > 3.0) | (VALUES.CRD_DELTA_H < -3.0))
{
    if (VALUES.CRD_DELTA_H > 1.0) //cm ziel - ist
        DEFLECTOR_UP();

    else if (VALUES.CRD_DELTA_H < 1.0) //cm    ziel - ist
        DEFLECTOR_DOWN();

    else DEFLECTOR_STOP();
}

// Keeps the display updated with current wind- and deflector conditions
    lcd_clear;
    lcd_send_float (VALUES.Alpha);
    lcd_string (" ");
    lcd_send_float (VALUES.Beta);
    lcd_string (" ");
    lcd_send_float (MPH);
    lcd_setcursor (0,2);
    lcd_send_float (VALUES.CRD_HEIGHT);
    lcd_string (" ");
    lcd_send_float (VALUES.CRD_TARGET);
    lcd_string (" ");
    lcd_send_float (VALUES.CRD_DELTA_H);
} //end void optimum_position(void)

```

```

void DEFLECTOR_UP (void)
{
if (VALUES.CRD_HEIGHT <= IR_upper_limit)
    {
        HB_PORT &= ~(1<<HB_IN_B);
        HB_PORT |= ((1<<HB_IN_A) | (1<<HB_EN) | (1<<HB_PWM));
    }
else DEFLECTOR_STOP();
}

/*
#define HB_IN_A      PB7          DOWN
#define HB_IN_B      PB4          UP
#define HB_EN        PB5          ENABLE
#define HB_PWM       PB6          PWM
#define HB_PORT      PORTB
*/

void DEFLECTOR_DOWN (void)
{
if      (VALUES.CRD_HEIGHT >= IR_lower_limit)
    {
        HB_PORT &= ~(1<<HB_IN_A);
        HB_PORT |= ((1<<HB_IN_B) | (1<<HB_EN) | (1<<HB_PWM));
    }
else DEFLECTOR_STOP();
}

void DEFLECTOR_STOP(void)
{
HB_PORT &= ~(1<<HB_IN_A | 1<<HB_IN_B | 1<<HB_EN | 1<<HB_PWM);
}

// this function is for smoothing out the IR Sensor reads.
// The ADC takes 10 reads of the ADC in about 0.3 ms.
// The read values get sorted from low to high and only the medium are averaged to an actual
read.
void sortme (void)
{
for (uint8_t n=0; n<5; ++n)
    {
        for (uint8_t i=0; i<(5-n-1); ++i)
            {
                if (VALUES.SORTME[i]> VALUES.SORTME[i+1])
                    {
                        temp_int16 = VALUES.SORTME[i];
                        VALUES.SORTME[i] = VALUES.SORTME[i+1];
                        VALUES.SORTME[i+1] = temp_int16;
                    }
            }
    }
VALUES.IR = (( VALUES.SORTME[1] + VALUES.SORTME[2] + VALUES.SORTME[3] ) / 3);
}

void lcd_send_float (double f)
{
char *buffer [11];
int16_t temp1 = (int) f;
int8_t n ;
sprintf (buffer, "%i.", temp1);
lcd_string (buffer);
for (n=0; n<1; n++)
    {
        f = f - (long double) temp1;
        f=f*10; // 1000 <>10
        if (f<0)

```

```

        f = f*(-1);
        templ= (int) f;
        sprintf (buffer, "%i", templ);
        lcd_string (buffer);
    }
}

// == the following lines are part of an open library for LCD displays based on HD44780
// display controllers. The display was used for debugging only. ==
/*****
/*****
// Ansteuerung eines HD44780 kompatiblen LCD im 4-Bit-Interfacemodus
// http://www.mikrocontroller.net/articles/HD44780
// http://www.mikrocontroller.net/articles/AVR-GCC-Tutorial/LCD-Ansteuerung
//
// Die Pinbelegung ist über defines in lcd-routines.h einstellbar

#include <avr/io.h>
#include "lcd-routines.h"
#include <util/delay.h>

////////////////////////////////////////////////////////////////
// Erzeugt einen Enable-Puls
static void lcd_enable( void )
{
    LCD_PORT |= (1<<LCD_EN);    // Enable auf 1 setzen
    _delay_us( LCD_ENABLE_US ); // kurze Pause
    LCD_PORT &= ~(1<<LCD_EN);   // Enable auf 0 setzen
}

////////////////////////////////////////////////////////////////
// Sendet eine 4-bit Ausgabeoperation an das LCD
static void lcd_out( uint8_t data )
{
    data &= 0xF0;                // obere 4 Bit maskieren

    LCD_PORT &= ~(0xF0>>(4-LCD_DB)); // Maske löschen
    LCD_PORT |= (data>>(4-LCD_DB)); // Bits setzen
    lcd_enable();
}

////////////////////////////////////////////////////////////////
// Initialisierung: muss ganz am Anfang des Programms aufgerufen werden.
void lcd_init( void )
{
    // verwendete Pins auf Ausgang schalten
    uint8_t pins = (0x0F << LCD_DB) | // 4 Datenleitungen
                  (1<<LCD_RS) |      // R/S Leitung
                  (1<<LCD_EN);       // Enable Leitung

    LCD_DDR |= pins;

    // initial alle Ausgänge auf Null
    LCD_PORT &= ~pins;

    // warten auf die Bereitschaft des LCD
    _delay_ms( LCD_BOOTUP_MS );

    // Soft-Reset muss 3mal hintereinander gesendet werden zur Initialisierung
    lcd_out( LCD_SOFT_RESET );
    _delay_ms( LCD_SOFT_RESET_MS1 );

    lcd_enable();
    _delay_ms( LCD_SOFT_RESET_MS2 );

    lcd_enable();
    _delay_ms( LCD_SOFT_RESET_MS3 );

    // 4-bit Modus aktivieren
    lcd_out( LCD_SET_FUNCTION |
            LCD_FUNCTION_4BIT );
    _delay_ms( LCD_SET_4BITMODE_MS );

    // 4-bit Modus / 2 Zeilen / 5x7
    lcd_command( LCD_SET_FUNCTION |

```

```

        LCD_FUNCTION_4BIT |
        LCD_FUNCTION_2LINE |
        LCD_FUNCTION_5X7 );

// Display ein / Cursor aus / Blinken aus
lcd_command( LCD_SET_DISPLAY |
             LCD_DISPLAY_ON |
             LCD_CURSOR_OFF |
             LCD_BLINKING_OFF);
// Cursor inkrement / kein Scrollen
lcd_command( LCD_SET_ENTRY |
             LCD_ENTRY_INCREASE |
             LCD_ENTRY_NOSHIFT );

    lcd_clear();
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Sendet ein Datenbyte an das LCD
void lcd_data( uint8_t data )
{
    LCD_PORT |= (1<<LCD_RS);    // RS auf 1 setzen

    lcd_out( data );           // zuerst die oberen,
    lcd_out( data<<4 );       // dann die unteren 4 Bit senden

    _delay_us( LCD_WRITEDATA_US );
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Sendet einen Befehl an das LCD
void lcd_command( uint8_t data )
{
    LCD_PORT &= ~(1<<LCD_RS);    // RS auf 0 setzen

    lcd_out( data );           // zuerst die oberen,
    lcd_out( data<<4 );       // dann die unteren 4 Bit senden

    _delay_us( LCD_COMMAND_US );
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Sendet den Befehl zur Löschung des Displays
void lcd_clear( void )
{
    lcd_command( LCD_CLEAR_DISPLAY );
    _delay_ms( LCD_CLEAR_DISPLAY_MS );
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Sendet den Befehl: Cursor Home
void lcd_home( void )
{
    lcd_command( LCD_CURSOR_HOME );
    _delay_ms( LCD_CURSOR_HOME_MS );
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Setzt den Cursor in Spalte x (0..15) Zeile y (1..4)
void lcd_setcursor( uint8_t x, uint8_t y )
{
    uint8_t data;

    switch (y)
    {
        case 1:    // 1. Zeile
            data = LCD_SET_DDADR + LCD_DDADR_LINE1 + x;
            break;

        case 2:    // 2. Zeile
            data = LCD_SET_DDADR + LCD_DDADR_LINE2 + x;
            break;

        case 3:    // 3. Zeile
            data = LCD_SET_DDADR + LCD_DDADR_LINE3 + x;

```

```

        break;

    case 4:    // 4. Zeile
        data = LCD_SET_DDADR + LCD_DDADR_LINE4 + x;
        break;

    default:
        return; // für den Fall einer falschen Zeile
}

lcd_command( data );
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Schreibt einen String auf das LCD

void lcd_string( const char *data )
{
    while( *data != '\0' )
        lcd_data( *data++ );
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Schreibt ein Zeichen in den Character Generator RAM

void lcd_generatechar( uint8_t code, const uint8_t *data )
{
    // Startposition des Zeichens einstellen
    lcd_command( LCD_SET.CGADR | (code<<3) );

    // Bitmuster übertragen
    for ( uint8_t i=0; i<8; i++ )
    {
        lcd_data( data[i] );
    }
}

```

CRD_config.h

```
/* *****
* Author: Jan Berkenbusch
* Controller: Atmel AT90CAN128 on CRD Board v6.0
* Dev Software: AVR Studio 4.19 (2011) with WinAVR Library (RC Oct/2010)
*
* Subject: Controller for Automatic Cab Roof Deflector for trucks
*
* CRD Config File for CRD_v60.c
*
* Content:
  - Implementation of five hole probe calibration parameters
  - Definition of deflector geometry
  - Threshold values for software end switch
  - Equations that describe the surfaces for CRD optimum position calculation
  -
***** */

//PS1
//Set of 5HP calibration parameters for pitch
#define PS1_pitch_a0 -7.4780
#define PS1_pitch_a1 -140.02
#define PS1_pitch_a2 1.6198
#define PS1_pitch_a3 380.76
#define PS1_pitch_a4 -0.6431
#define PS1_pitch_a5 -23.259
#define PS1_pitch_a6 62.475
#define PS1_pitch_a7 11.629
#define PS1_pitch_a8 -507.42
#define PS1_pitch_a9 0.5384
#define PS1_pitch_a10 -7.8622
#define PS1_pitch_a11 -39.266
#define PS1_pitch_a12 -1.2780
#define PS1_pitch_a13 227.77
#define PS1_pitch_a14 -0.1965
//accuracy pitch: 0.8844

//PS1
//Set of 5HP calibration parameters for yaw
#define PS1_yaw_a0 -34.743
#define PS1_yaw_a1 383.17
#define PS1_yaw_a2 8.9267
#define PS1_yaw_a3 -1205.9
#define PS1_yaw_a4 7.0635
#define PS1_yaw_a5 64.967
#define PS1_yaw_a6 -167.00
#define PS1_yaw_a7 -49.970
#define PS1_yaw_a8 1596.3
#define PS1_yaw_a9 -2.3112
#define PS1_yaw_a10 55.355
#define PS1_yaw_a11 126.12
#define PS1_yaw_a12 0.6970
#define PS1_yaw_a13 -785.03
#define PS1_yaw_a14 0.2337
//accuracy yaw: 0.7113

//PS1
//Set of 5HP calibration parameters for static
#define PS1_static_a0 1.8436
#define PS1_static_a1 -6.1882
#define PS1_static_a2 0.0454
#define PS1_static_a3 12.113
#define PS1_static_a4 -0.0866
#define PS1_static_a5 -0.6369
#define PS1_static_a6 2.0496
#define PS1_static_a7 0.6638
#define PS1_static_a8 -15.8834
#define PS1_static_a9 0.0172
#define PS1_static_a10 -0.6264
#define PS1_static_a11 -1.4913
#define PS1_static_a12 -0.0221
#define PS1_static_a13 7.9097
```

```

#define PS1_static_a14          -0.0030
//accuracy static: 0.9730

//PS1
//Set of 5HP calibration parameters for total
#define PS1_total_a0           -0.3977
#define PS1_total_a1            0.7382
#define PS1_total_a2            0.0887
#define PS1_total_a3            1.5033
#define PS1_total_a4           -0.1548
#define PS1_total_a5           -0.5966
#define PS1_total_a6            0.8134
#define PS1_total_a7            0.3077
#define PS1_total_a8           -4.1015
#define PS1_total_a9           -0.0111
#define PS1_total_a10          -0.4755
#define PS1_total_a11          -0.3240
#define PS1_total_a12           0.0335
#define PS1_total_a13           2.2998
#define PS1_total_a14           0.1010
//accuracy total: 0.9702

//PS2
//Set of 5HP calibration parameters for pitch
#define PS2_pitch_a0           -0.1672
#define PS2_pitch_a1            4.1847
#define PS2_pitch_a2           -0.9292
#define PS2_pitch_a3            2.3388
#define PS2_pitch_a4            34.652
#define PS2_pitch_a5           -53.882
#define PS2_pitch_a6           -9.6557
#define PS2_pitch_a7            11.776
#define PS2_pitch_a8           -4.6777
#define PS2_pitch_a9           -48.616
#define PS2_pitch_a10           1.5559
#define PS2_pitch_a11           14.584
#define PS2_pitch_a12          -12.026
#define PS2_pitch_a13            0.4634
#define PS2_pitch_a14           26.918
//accuracy pitch: 0.9534

//PS2
//Set of 5HP calibration parameters for yaw
#define PS2_yaw_a0              12.314
#define PS2_yaw_a1              3.8515
#define PS2_yaw_a2              68.881
#define PS2_yaw_a3             -0.5018
#define PS2_yaw_a4             -145.95
#define PS2_yaw_a5             -13.869
#define PS2_yaw_a6             -5.1924
#define PS2_yaw_a7              32.882
#define PS2_yaw_a8             -2.7119
#define PS2_yaw_a9              189.05
#define PS2_yaw_a10            -5.4480
#define PS2_yaw_a11              3.3510
#define PS2_yaw_a12            -21.090
#define PS2_yaw_a13              0.9802
#define PS2_yaw_a14            -69.913
//accuracy yaw: 0.9301

//PS2
//Set of 5HP calibration parameters for static
#define PS2_static_a0           2.2888
#define PS2_static_a1            0.0503
#define PS2_static_a2           -7.9608
#define PS2_static_a3           -0.0821
#define PS2_static_a4           13.472
#define PS2_static_a5           -0.0344
#define PS2_static_a6            0.3182
#define PS2_static_a7           -0.1219
#define PS2_static_a8           -0.0187
#define PS2_static_a9          -12.729
#define PS2_static_a10          -0.0795
#define PS2_static_a11           0.0070

```

```

#define PS2_static_a12          0.1114
#define PS2_static_a13          0.0033
#define PS2_static_a14          4.2727
//accuracy static: 0.9631

//PS2
//Set of 5HP calibration parameters for total
#define PS2_total_a0            -0.4271
#define PS2_total_a1            -0.0205
#define PS2_total_a2            0.9630
#define PS2_total_a3            -0.0266
#define PS2_total_a4            0.7005
#define PS2_total_a5            -0.2337
#define PS2_total_a6            -0.5041
#define PS2_total_a7            1.0253
#define PS2_total_a8            -0.0457
#define PS2_total_a9            -2.7005
#define PS2_total_a10           0.3262
#define PS2_total_a11           0.0660
#define PS2_total_a12           -0.9085
#define PS2_total_a13           0.0176
#define PS2_total_a14           1.2982
//accuracy total: 0.9212

//PS3
//Set of 5HP calibration parameters for pitch
#define PS3_pitch_a0            14.888
#define PS3_pitch_a1            -78.082
#define PS3_pitch_a2            10.595
#define PS3_pitch_a3            -214.36
#define PS3_pitch_a4            -2.9218
#define PS3_pitch_a5            51.918
#define PS3_pitch_a6            91.457
#define PS3_pitch_a7            -7.246
#define PS3_pitch_a8            -311.77
#define PS3_pitch_a9            -2.3119
#define PS3_pitch_a10           -4.9910
#define PS3_pitch_a11           56.161
#define PS3_pitch_a12           -3.8576
#define PS3_pitch_a13           -149.91
#define PS3_pitch_a14           -0.3026
//accuracy pitch: 0.9441

//PS3
//Set of 5HP calibration parameters for yaw
#define PS3_yaw_a0              9.3471
#define PS3_yaw_a1              129.48
#define PS3_yaw_a2              -13.367
#define PS3_yaw_a3              351.84
#define PS3_yaw_a4              10.046
#define PS3_yaw_a5              -156.33
#define PS3_yaw_a6              -261.70
#define PS3_yaw_a7              21.517
#define PS3_yaw_a8              365.23
#define PS3_yaw_a9              -0.7826
#define PS3_yaw_a10             15.855
#define PS3_yaw_a11             -146.65
#define PS3_yaw_a12             2.3714
#define PS3_yaw_a13             125.70
#define PS3_yaw_a14             -0.7562
//accuracy yaw: 0.9700

//PS3
//Set of 5HP calibration parameters for static
#define PS3_static_a0           2.3002
#define PS3_static_a1           8.4418
#define PS3_static_a2           0.1830
#define PS3_static_a3           16.413
#define PS3_static_a4           0.0987
#define PS3_static_a5           0.1892
#define PS3_static_a6           0.1717
#define PS3_static_a7           -0.1659
#define PS3_static_a8           18.265
#define PS3_static_a9           -0.1038

```

```

#define PS3_static_a10      -0.1815
#define PS3_static_a11      0.2409
#define PS3_static_a12      -0.1234
#define PS3_static_a13      7.9327
#define PS3_static_a14      -0.0290
//accuracy static: 0.9635

//PS3
//Set of 5HP calibration parameters for total
#define PS3_total_a0        -0.8690
#define PS3_total_a1        -6.9112
#define PS3_total_a2         0.2467
#define PS3_total_a3       -20.395
#define PS3_total_a4        -0.2634
#define PS3_total_a5         2.1813
#define PS3_total_a6         4.4046
#define PS3_total_a7        -0.7103
#define PS3_total_a8       -24.329
#define PS3_total_a9        -0.0638
#define PS3_total_a10       -0.6089
#define PS3_total_a11         2.6885
#define PS3_total_a12       -0.0638
#define PS3_total_a13       -10.234
#define PS3_total_a14       -0.0069
//accuracy total: 0.7422

//PS4
//Set of 5HP calibration parameters for pitch
#define PS4_pitch_a0         -6.4497
#define PS4_pitch_a1          0.8049
#define PS4_pitch_a2        -74.553
#define PS4_pitch_a3         -6.7172
#define PS4_pitch_a4       -299.77
#define PS4_pitch_a5         64.761
#define PS4_pitch_a6        -38.036
#define PS4_pitch_a7         61.637
#define PS4_pitch_a8        -0.5949
#define PS4_pitch_a9       -396.82
#define PS4_pitch_a10       -28.722
#define PS4_pitch_a11        -7.4823
#define PS4_pitch_a12         42.983
#define PS4_pitch_a13        -1.2536
#define PS4_pitch_a14       -164.96
//accuracy pitch: 0.9688

//PS4
//Set of 5HP calibration parameters for yaw
#define PS4_yaw_a0           -17.379
#define PS4_yaw_a1            2.2395
#define PS4_yaw_a2            6.8369
#define PS4_yaw_a3           -1.7426
#define PS4_yaw_a4           -30.307
#define PS4_yaw_a5           -5.0776
#define PS4_yaw_a6           -20.946
#define PS4_yaw_a7          -14.6832
#define PS4_yaw_a8           -0.7876
#define PS4_yaw_a9            52.991
#define PS4_yaw_a10          -12.595
#define PS4_yaw_a11          -0.8519
#define PS4_yaw_a12          -6.0930
#define PS4_yaw_a13          -0.8125
#define PS4_yaw_a14           61.621
//accuracy yaw: 0.9120

//PS4
//Set of 5HP calibration parameters for static
#define PS4_static_a0         1.0470
#define PS4_static_a1         0.3515
#define PS4_static_a2        -1.7409
#define PS4_static_a3        -0.3018
#define PS4_static_a4       -14.765
#define PS4_static_a5         2.3481
#define PS4_static_a6        -1.1733
#define PS4_static_a7         4.2419

```

```

#define PS4_static_a8          -0.0305
#define PS4_static_a9         -19.417
#define PS4_static_a10        -0.9271
#define PS4_static_a11        -0.0316
#define PS4_static_a12         2.3790
#define PS4_static_a13         0.0126
#define PS4_static_a14        -7.8853
//accuracy static: 0.9320

//PS4
//Set of 5HP calibration parameters for total
#define PS4_total_a0          -0.4001
#define PS4_total_a1           0.0525
#define PS4_total_a2          -1.8167
#define PS4_total_a3          -0.0255
#define PS4_total_a4          -4.5453
#define PS4_total_a5           1.0172
#define PS4_total_a6           0.3039
#define PS4_total_a7           2.0429
#define PS4_total_a8           0.0320
#define PS4_total_a9          -6.4020
#define PS4_total_a10          0.1771
#define PS4_total_a11          0.0383
#define PS4_total_a12          1.0871
#define PS4_total_a13          0.0097
#define PS4_total_a14          -3.5651
//accuracy total: 0.7851

//PS5
//Set of 5HP calibration parameters for pitch
#define PS5_pitch_a0          -7.3199
#define PS5_pitch_a1          -11.296
#define PS5_pitch_a2          -3.3788
#define PS5_pitch_a3           6.9392
#define PS5_pitch_a4           4.0673
#define PS5_pitch_a5          -0.8078
#define PS5_pitch_a6           0.8611
#define PS5_pitch_a7           0.2196
#define PS5_pitch_a8           0.4604
#define PS5_pitch_a9           1.6814
#define PS5_pitch_a10          -1.6895
#define PS5_pitch_a11          -0.0141
#define PS5_pitch_a12           0.1998
#define PS5_pitch_a13          -1.4920
#define PS5_pitch_a14          -0.4371
//accuracy pitch: 0.8416

//PS5
//Set of 5HP calibration parameters for yaw
#define PS5_yaw_a0             1.9431
#define PS5_yaw_a1             0.7818
#define PS5_yaw_a2             9.5566
#define PS5_yaw_a3             -1.7473
#define PS5_yaw_a4             -2.0807
#define PS5_yaw_a5             0.4104
#define PS5_yaw_a6             0.0696
#define PS5_yaw_a7             0.0467
#define PS5_yaw_a8             -0.0129
#define PS5_yaw_a9             0.0510
#define PS5_yaw_a10            0.0393
#define PS5_yaw_a11            -0.3343
#define PS5_yaw_a12            -0.4451
#define PS5_yaw_a13            0.4842
#define PS5_yaw_a14            0.7851
//accuracy yaw: 0.9786

//PS5
//Set of 5HP calibration parameters for static
#define PS5_static_a0           0.7636
#define PS5_static_a1          -0.2388
#define PS5_static_a2          -0.1338
#define PS5_static_a3           0.3105
#define PS5_static_a4           0.1840
#define PS5_static_a5           0.0117

```

```

#define PS5_static_a6          0.0601
#define PS5_static_a7          0.0703
#define PS5_static_a8          0.0819
#define PS5_static_a9          0.0657
#define PS5_static_a10         -0.1007
#define PS5_static_a11         0.0248
#define PS5_static_a12         -0.10302
#define PS5_static_a13         -0.0970
#define PS5_static_a14         -0.0568
//accuracy static: 0.2369

//PS5
//Set of 5HP calibration parameters for total
#define PS5_total_a0           -0.0621
#define PS5_total_a1           -0.0257
#define PS5_total_a2           -0.0909
#define PS5_total_a3           0.0340
#define PS5_total_a4           -0.0515
#define PS5_total_a5           -0.0550
#define PS5_total_a6           0.0014
#define PS5_total_a7           -0.0046
#define PS5_total_a8           0.0098
#define PS5_total_a9           0.0301
#define PS5_total_a10          -0.0580
#define PS5_total_a11          0.0159
#define PS5_total_a12          0.0219
#define PS5_total_a13          -0.0254
#define PS5_total_a14          0.0193
//accuracy total: 0.7067

//define "zero" values for ACD reads
#define Ref_Volts_PS1          900 //ADC TICS
#define Ref_Volts_PS2          900 //ADC TICS
#define Ref_Volts_PS3          900 //ADC TICS
#define Ref_Volts_PS4          900 //ADC TICS
#define Ref_Volts_PS5          900 //ADC TICS

// positioning tolerance for CRD height in cm
#define Positioning_window     10 //in cm
#define Roof_height            400 //in cm

// density of air in kg/m3
#define rho_air 1.225

//cab dimensions
#define IR_upper_limit(78.0 + cab_roof_height) //in cm
#define IR_lower_limit(62.0 + cab_roof_height) //in cm // was 59.0
#define cab_roof_height        400.0 //in cm

//declaration of HBridge Output Ports
#define HB_IN_A                PB7
#define HB_IN_B                PB4
#define HB_EN                  PB5
#define HB_PWM                 PB6
#define HB_PORT                PORTB

/*selecting height in steps. 4.2m - 4.6m - 4.8m
4.2      0      10      20      30      alpha
#=====#=====#=====#=====#
|      |      |      |      |      |
40      |      4.4      |      4.4      |      4.4      |      4.4      |
|      |      |      |      |      |
#=====# 1  ===== 3  ===== 5  =====#  > height
|      |      |      |      |      |
50      |      4.4      |      4.4      |      4.4      |      4.4      |
|      |      |      |      |      |
#=====# 2  ===== 4  ===== 6  =====#
|      |      |      |      |      |
56      |      4.4      |      4.4      |      4.4      |      4.4      |
|      |      |      |      |      |
#=====#=====#=====#=====#

```

```

speed
4.6 0 10 20 30 alpha
#-----#-----#-----#-----#
| 4.5 | 4.7 | 4.8 | 4.8 |
#===== 1 ===== 3 ===== 5 =====# > height
| 4.5 | 4.7 | 4.8 | 4.8 |
#===== 2 ===== 4 ===== 6 =====#
| 4.5 | 4.7 | 4.8 | 4.8 |
#-----#-----#-----#-----#
speed
4.8 0 10 20 30 alpha
#-----#-----#-----#-----#
| 4.6 | 4.7 | 4.9 | 4.9 |
#===== 1 ===== 3 ===== 5 =====# > height
| 4.6 | 4.7 | 4.9 | 4.9 |
#===== 2 ===== 4 ===== 6 =====#
| 4.6 | 4.7 | 4.9 | 4.9 |
#-----#-----#-----#-----#
speed
*/

```

```

// Equations that describe the surfaces for CRD optimum position calculation
#define eq4_2__1 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm
#define eq4_2__2 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm
#define eq4_2__3 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm
#define eq4_2__4 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm
#define eq4_2__5 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm
#define eq4_2__6 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm

#define eq4_6__1 (VALUES.CRD_REFERENCE - 10) + (2 * VALUES.Beta)
// h = h_ref -10 + 2*alpha in cm
#define eq4_6__2 (VALUES.CRD_REFERENCE - 10) + (2 * VALUES.Beta)
// h = h_ref -10 + 2*alpha in cm
#define eq4_6__3 (VALUES.CRD_REFERENCE + 10) + (VALUES.Beta)
// h = h_ref +10 + alpha in cm
#define eq4_6__4 (VALUES.CRD_REFERENCE + 10) + (VALUES.Beta)
// h = h_ref +10 + alpha in cm
#define eq4_6__5 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm
#define eq4_6__6 (VALUES.CRD_REFERENCE + 20)
// h = h_ref +20 in cm

#define eq4_8__1 (VALUES.CRD_REFERENCE - 20) + (1 * VALUES.Beta)
// h = h_ref -20 + 1 * alpha in cm
#define eq4_8__2 (VALUES.CRD_REFERENCE - 20) + (1 * VALUES.Beta)
// h = h_ref -20 + 1 * alpha in cm
#define eq4_8__3 (VALUES.CRD_REFERENCE - 30) + (2 * VALUES.Beta)
// h = h_ref -30 + 2 * alpha in cm
#define eq4_8__4 (VALUES.CRD_REFERENCE - 30) + (2 * VALUES.Beta)
// h = h_ref -30 + 2 * alpha in cm
#define eq4_8__5 (VALUES.CRD_REFERENCE + 10)
// h = h_ref +10 in cm
#define eq4_8__6 (VALUES.CRD_REFERENCE + 10)
// h = h_ref +10 in cm

```

LCD_routines.h

```
// Ansteuerung eines HD44780 kompatiblen LCD im 4-Bit-Interfacemodus
// http://www.mikrocontroller.net/articles/AVR-GCC-Tutorial/LCD-Ansteuerung
//

#ifndef LCD_ROUTINES_H
#define LCD_ROUTINES_H

////////////////////////////////////
// Hier die verwendete Taktfrequenz in Hz eintragen, wichtig!

#ifndef F_CPU
#define F_CPU 8000000
#endif

////////////////////////////////////
// Pinbelegung für das LCD, an verwendete Pins anpassen
// Alle LCD Pins müssen an einem Port angeschlossen sein und die 4
// Datenleitungen müssen auf aufeinanderfolgenden Pins liegen

// LCD DB4-DB7 <--> PORTD Bit PD4-PD7
#define LCD_PORT PORTC
#define LCD_DDR DDRC
#define LCD_DB PC0

// LCD RS <--> PORTD Bit PD4 (RS: 1=Data, 0=Command)
#define LCD_RS PC4

// LCD EN <--> PORTD Bit PD5 (EN: 1-Impuls für Daten)
#define LCD_EN PC5

////////////////////////////////////
// LCD Ausführungszeiten (MS=Millisekunden, US=Mikrosekunden)

#define LCD_BOOTUP_MS 15
#define LCD_ENABLE_US 30 //20
#define LCD_WRITEDATA_US 46
#define LCD_COMMAND_US 42 //42

#define LCD_SOFT_RESET_MS1 5
#define LCD_SOFT_RESET_MS2 1
#define LCD_SOFT_RESET_MS3 1
#define LCD_SET_4BITMODE_MS 5

#define LCD_CLEAR_DISPLAY_MS 4 //2
#define LCD_CURSOR_HOME_MS 2

////////////////////////////////////
// Zeilendefinitionen des verwendeten LCD
// Die Einträge hier sollten für ein LCD mit einer Zeilenlänge von 16 Zeichen passen
// Bei anderen Zeilenlängen müssen diese Einträge angepasst werden

#define LCD_DDADR_LINE1 0x00
#define LCD_DDADR_LINE2 0x40
#define LCD_DDADR_LINE3 0x10
#define LCD_DDADR_LINE4 0x50

////////////////////////////////////
// Initialisierung: muss ganz am Anfang des Programms aufgerufen werden.
void lcd_init( void );

////////////////////////////////////
// LCD löschen
void lcd_clear( void );

////////////////////////////////////
// Cursor in die 1. Zeile, 0-te Spalte
void lcd_home( void );

////////////////////////////////////
// Cursor an eine beliebige Position
void lcd_setcursor( uint8_t spalte, uint8_t zeile );
```

```

////////////////////////////////////
// Ausgabe eines einzelnen Zeichens an der aktuellen Cursorposition
void lcd_data( uint8_t data );

////////////////////////////////////
// Ausgabe eines Strings an der aktuellen Cursorposition
void lcd_string( const char *data );

////////////////////////////////////
// Definition eines benutzerdefinierten Sonderzeichens.
// data muss auf ein Array[8] mit den Zeilencodes des zu definierenden Zeichens
// zeigen
void lcd_generatechar( uint8_t code, const uint8_t *data );

////////////////////////////////////
// Ausgabe eines Kommandos an das LCD.
void lcd_command( uint8_t data );

////////////////////////////////////
// LCD Befehle und Argumente.
// Zur Verwendung in lcd_command

// Clear Display ----- 0b00000001
#define LCD_CLEAR_DISPLAY      0x01

// Cursor Home ----- 0b0000001x
#define LCD_CURSOR_HOME      0x02

// Set Entry Mode ----- 0b000001xx
#define LCD_SET_ENTRY        0x04

#define LCD_ENTRY_DECREASE    0x00
#define LCD_ENTRY_INCREASE    0x02
#define LCD_ENTRY_NOSHIFT     0x00
#define LCD_ENTRY_SHIFT      0x01

// Set Display ----- 0b00001xxx
#define LCD_SET_DISPLAY      0x08

#define LCD_DISPLAY_OFF      0x00
#define LCD_DISPLAY_ON       0x04
#define LCD_CURSOR_OFF      0x00
#define LCD_CURSOR_ON       0x02
#define LCD_BLINKING_OFF    0x00
#define LCD_BLINKING_ON     0x01

// Set Shift ----- 0b0001xxxx
#define LCD_SET_SHIFT        0x10

#define LCD_CURSOR_MOVE      0x00
#define LCD_DISPLAY_SHIFT    0x08
#define LCD_SHIFT_LEFT      0x00
#define LCD_SHIFT_RIGHT     0x04

// Set Function ----- 0b001xxxxx
#define LCD_SET_FUNCTION     0x20

#define LCD_FUNCTION_4BIT    0x00
#define LCD_FUNCTION_8BIT    0x10
#define LCD_FUNCTION_1LINE   0x00
#define LCD_FUNCTION_2LINE   0x08
#define LCD_FUNCTION_5X7     0x00
#define LCD_FUNCTION_5X10    0x04

#define LCD_SOFT_RESET      0x30

// Set CG RAM Address ----- 0b01xxxxxx (Character Generator RAM)
#define LCD_SET_CGADR        0x40

#define LCD_GC_CHAR0        0
#define LCD_GC_CHAR1        1
#define LCD_GC_CHAR2        2
#define LCD_GC_CHAR3        3

```

```
#define LCD_GC_CHAR4      4
#define LCD_GC_CHAR5      5
#define LCD_GC_CHAR6      6
#define LCD_GC_CHAR7      7

// Set DD RAM Address ----- 0b1xxxxxxx (Display Data RAM)
#define LCD_SET_DDADR      0x80

#endif
```

A5 DVD with CFD results files

stick DVD sleeve here