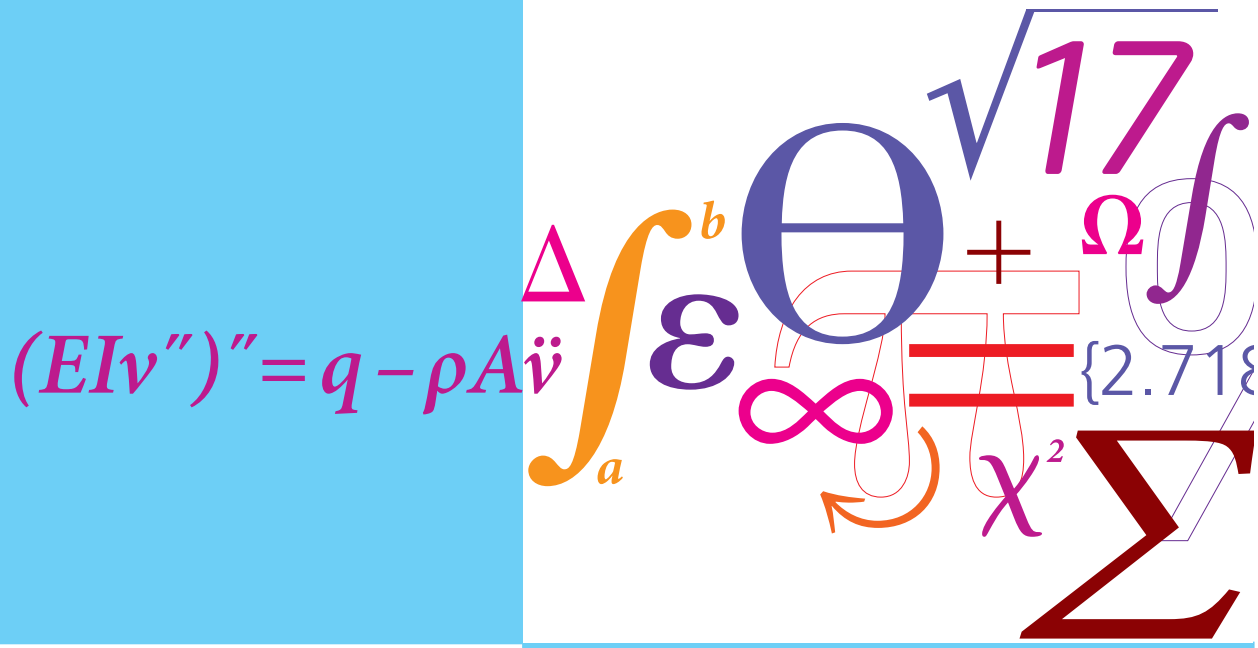
Measurement of turbulence in an IC engine with hot-wire anemometry

Master Thesis



Rasmus Katkjær
March 2016

DTU Mechanical Engineering
Department of Mechanical Engineering

Measurement of turbulence in an IC engine with hot-wire anemometry

Master Thesis

Rasmus Katkjær - s103380

Supervisor
Anders Ivarson
Clara Marika Velte

13-03-2016

Abstract

The purpose of this study is to establish a basis for measurements of turbulent structures in an IC engine by using hot-wire anemometry. This was done via a simple pipe test, for obtaining knowledge and experience in the use of hot-wire anemometry, and later in an non-fired, 1 cylinder, 4 stroke SI IC Briggs and Stratton engine from a lawnmower.

The pipe test was carried out with a 1D hot-wire probe, in two positions downstream and throughout the pipe diameter. The test gave a good experience for the use of hot-wire anemometry, and by comparing the obtained result with established pipe results, it was concluded that the set-up was able to recreate trends and result equal to established results.

With the knowledge and experience from the pipe test, it was possible to make an engine test set-up. The engine test was performed with a 1D high-temperature probe, which was mounted in the spark plough hole. A fixture was designed and tested to make sure that the hot-wire support was not damaged or blown out of the engine under operation. Measurement was carried out at two positions, in two directions at three different engine speeds.
It was found, that the equipment used was able to create a voltage signal from the hot-wire with attached pulses from an inductive sensor, in order to determine the position of the piston. It was found that a maximum sampling frequency of $100kHz$ was not sufficient for resolving the Kolmogorov scales in the flow, and a low-pass filter of $10kHz$ was to small.

It was concluded that a careful data processing is necessary for obtaining good results. A simple simulation for estimating gas pressure and temperature was carried out and was found sufficient for this experiment, but for further work, a more advanced simulating or a hot-wire with an integrated thermocouple can be used.
A temperature correction has to be carried out, due to the high pressure and temperature differences in the engine. A correcting which takes conduction to the wire prongs into account was preferred, but was neglected in this study. It is recompensed to make the separation of the fluctuations from the velocity signal with a high-pass filter.
An issue regarding dividing the signal of an engine cycle into intervals was encountered. If a good resolution of the cycle was made in order to see the development during the cycle, the intervals was to shot and was affected from windowing. A good compromise or an other way of applying the intervals should be looked into in further work.

Preface

This is a master thesis from the master program *Engineering design and applied mechanics* at the Technical University of Denmark.

I would like to thank Dr. Martin Thalbitzer Andersen and F. Gökhan Ergin from DanTec Dynamics for guidance in the process of selecting the hot-wire and the appurtenant equipment.
I would like to thank Laboratory engineer Benny Edelsten from DTU MEK for the demonstration of the hot-wire set-up and calibration regarding the pipe test. Thanks to the DTU Workshop in building 414 for guidance, production and ordering of parts for the engine test.
Thank you to PhD student Mads Carsten Jørgensen and postdoc Claus Suldrup Nielsen from DTU MEK for supervising and guidance regarding the engine test.

A big thank to associated professor Clara Marika Velte from DTU MEK for supervising. And a special thank to my supervisor associated professor Anders Ivarsson from DTU MEK for guidance and helping me throughout the entire project.

Contents

Nomenclature

Symbol

A	Cross sectional area, area under curve, calibration constant	$[m^2]$, $[m^2]$, $[-]$
A_w	Wire surface area	$[m^2]$
a	Overheat ratio	$[-]$
B	Cylinder bore, calibration constant	$[m]$, $[-]$
C	Constant	$[-]$
CF	Correction factor	$[-]$
D	Pipe diameter, dimension	$[m]$, $[-]$
$D_{support}$	Hot-wire support diameter	$[m]$
D_w	Wire diameter	$[m]$
E	Voltage, signal energy	$[V]$, $[\frac{m^2}{s}]$
f	Frequency	$[Hz]$
f_c	Critical frequency	$[Hz]$
$f_{cut-off}$	Cut-off frequency	$[Hz]$
f_{int}	Integral length scale expressed in frequency	$[Hz]$
f_{kol}	Kolmogorov length scale expressed in frequency	$[Hz]$
f_{max}	Maximum frequency	$[Hz]$
f_λ	Taylor length scale expressed in frequency	$[Hz]$
g	Gravitational acceleration	$[\frac{m}{s^2}]$
h	Convection heat transfer coefficient	$[\frac{W}{m^2 \cdot K}]$
I	Current	$[A]$
k_cor	Corrected thermal conductivity	$[\frac{W}{m \cdot K}]$
k_f	Thermal conductivity of fluid	$[\frac{W}{m \cdot K}]$
k_0	Thermal conductivity of fluid at reference temperature	$[\frac{W}{m \cdot K}]$
L	Connecting rod	$[m]$
L_{int}	Integral length scale	$[m]$
L_{kol}	Kolmogorov length scale	$[m]$
L_w	Wire length	$[m]$
N	Number of samples	$[-]$
N_{block}	Number of samples in a block	$[-]$
N_{total}	Total number of samples	$[-]$
n	Calibration constant	$[-]$
p	Pressure	$[Pa]$
p_{atm}	Atmospheric pressure	$[Pa]$
p_0	Pressure at reference temperature	$[Pa]$
$\dot{q}$	convection heat transfer flux	$[\frac{W}{m^2}]$
R	Resistance, crank shaft arm length, normalized autocorrelation function	$[\Omega]$, $[m]$, $[-]$
R_{total}	Total bridge resistance	$[\Omega]$
R_{spec}	specific gas constant	$[\frac{J}{kg \cdot K}]$
R_w	Wire resistance	$[\Omega]$
R_0	Wire resistance at reference temperature	$[\Omega]$
r	Autocorrelation function	$[-]$
S	Stroke	$[m]$

T	Temperature, sampling time	$[K]$, $[s]$
T_{atm}	Atmospheric temperature	$[K]$
T_g	Gas temperature	$[K]$
T_{int}	Integral time scale	$[s]$
T_{kol}	Kolmogorov time scale	$[s]$
T_w	Wire temperature	$[K]$
T_0	Wire temperature at reference temperature	$[K]$
t	Time	$[s]$
t_{block}	Block time	$[s]$
t_{total}	Total recording time	$[s]$
V	Velocity, volume	$[\frac{m}{s}]$, $[m^3]$
V_c	Compression volume	$[m^3]$
V_{cl}	Center line velocity	$[\frac{m}{s}]$
V_d	Displacement volume	$[m^3]$
V_f	Velocity fluctuations	$[\frac{m}{s}]$
V_{std}	Standard deviation of velocity signal	$[\frac{m}{s}]$
V_{var}	Variance of velocity signal	$[\frac{m}{s}]$
$\bar{V}$	Mean velocity	$[\frac{m}{s}]$
$\hat{V}$	Output of a FFT	$[\frac{m}{s}]$
$\hat{V}^*$	Complex output of a FFT	$[\frac{m}{s}]$
v	Velocity	$[\frac{m}{s}]$

Greek symbols

α	Temperature coefficient of resistance	$[\frac{\%}{^\circ C}]$
ε	Dissipation rate, compression ratio	$[-]$, $[-]$
γ	Specific heat ratio	$[-]$
κ	Wave number	$[\frac{1}{m}]$
λ	Taylor microscale	$[m]$
ν	Kinematic viscosity	$[\frac{m^2}{s}]$
μ	Dynamic viscosity	$[\frac{kg}{m \cdot s}]$
μ_{cor}	Corrected dynamic viscosity	$[\frac{kg}{m \cdot s}]$
μ_0	Kinematic viscosity at reference temperature	$[\frac{kg}{m \cdot s}]$
ρ	Density	$[\frac{kg}{m^3}]$
τ	Time step	$[s]$
θ	Crank angle	$[deg]$

Abbreviations

$ABDC$	After bottom dead center	
$ATDC$	After top dead center	
BDC	Bottom dead center	
$BTDC$	Before top dead center	
CAD	Crank angle degrees	
CCA	Constant current anemometry	
CTA	Constant temperature anemometry	
FFT	Fast Fourier transformation	
HWA	Hot-wore anemometry	
IC	Internal combustion	
$IFFT$	Inverse fast Fourier transformation	
int	Interval	
LDA	Laser dobbler anemometry	
Nu	Nusselt number	
PIV	Particle image velocimetry	
RAM	Random memory access	
Re	Reynolds number	
RTI	Relative turbulence intensity	
rpm	Revolutions per minute	$[\frac{Rev}{min}]$
SD	Sample duration	$[s]$
SI	Spark ignition	
SR	Sample rate	$[Hz]$
TDC	Top dead center	

1 Introduction

In the search for more efficient and environmental friendly engines, a variety of different initiatives to improve the engines have been launched. With more and more restrictions from governments and associations a more complete understanding of engine parameters have to be obtained. One of the areas which is important to understand is the combustion process.
The combustion process is among others depending on the mixing of the fuel and air. It has been shown that the movement and composition of turbulent structures of the air plays an import roll in order to control the combustion. In the perspective of gaining more knowledge about the early stages of the combustion process, the present study investigates and aims at understanding such turbulent structures..

Turbulent structures can be investigated through many different techniques. Common techniques used in experimental fluid mechanics are Laser Doppler Anemometry, Particle Image Velocimetry and Hot-Wire Anemometry. A pre-study has been carried out in order to determinate the pros and cons for each methods, showing that the method of Hot-Wire Anemometry is the one most suitable to the scale of this study. This technique was therefore applied.

The goal of this study is to obtain knowledge and experiences in the use of the Hot-Wire Anemometry technique and establish a basis for performing measurements of turbulent structures in an IC engine. The turbulent structures which are sought for are the integral time and length scales, the Taylor microscale and the Kolmogorov time and length scales. In order to get familiar with the Hot-Wire Anemometry technique, a pipe experiment was designed, and measurement of turbulent structures was performed. The pipe experiment was carried out in the main flow direction at two positions downstream and throughout the radius of the pipe. The advantage of a pipe experiment is that the pipe flow is a well defined flow, so obtained results can be compared to known behaviours.
With the experience from the pipe experiment, an engine experiment was then designed and preliminary measurement of turbulent structures was performed. The engine experiment is more complex than the pipe experiment due to varying temperature and pressure, a more multi directional flow and issues regarding the fast movements of the engine. The obtained results from the engine experiment are rather simple, but are important in order to gain knowledge of how to use Hot-Wire Anemometry in an engine. The data processing is an important part of the engine experiment and will be discussed throughout this report. The limitations and possible improvements regarding measurements in an engine will be discussed and presented, and a deeper analysis of different aspect of the data processing will be carried out.

2 Theory

In this chapter, the theory used in this study is presented. A presentation of hot-wires, turbulent structures and data processing can be found.

2.1 Hot-wire

In this study, the method used for measuring turbulence, is hot-wire anemometry (HWA). The hot-wire anemometer can measure fluid velocities which can be used to determine turbulent parameters. A hot-wire anemometer consists of a thin wire suspended between support needles, also called prongs. A sketch can be seen in Figure 1.

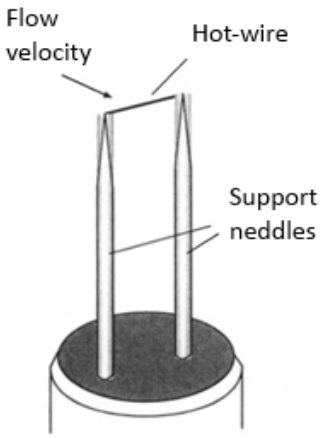


Figure 1: Sketch of 1D hot-wire probe

This is an example of a 1D hot-wire, but also 2D and 3D hot-wires are available. An electric current is sent trough the wire, which will lead to an increase of the wire temperature, through the resistance of the wire. The resistance of the wire is changing as a function of wire temperature.

$$R_w(T_w) = R_0(1 + \alpha(T_w - T_0)) \tag{1}$$

Where R_w is the wire resistance, T_w is the current wire temperature, T_0 is the start wire temperature, R_0 is the resistant at T_0 and α is the temperature coefficient of resistance for the wire. When the hot-wire is put in a flow, the colder fluid will cools down the wire, and thereby changes the wire resistance. The cooling effect dependent on the flow speed and fluid parameters. By investigating the energy budget, comparing the amount of electric energy put into the wire and the amount of thermal energy transferred to the flow, the flow velocity can be found.

The advantage of HWA is that the equipment is relatively inexpensive compared to other methods like laser doppler anemometry (LDA) and particle image velocimetry (PIV). At the same time HWA has a high frequency response, a continuous output signal and the system is able to measure in multiple dimensions, D. In engine applications, HWA is also relatively easy to install, again compared to LDA or PIV. These two methods are optical

techniques and therefore require optical access to the engine cylinder.
The disadvantage of HWA is that it interferes with the flow. This means that the flow can be affected by the hot-wire and also that the flow cannot contain particles which can damage or contaminate the wire. The wire itself is fragile because the wire has to be thin in order to have a good frequency response. In engine application, this means that measurements have to be carried out without combustion, so only motored results can be obtained. This is in most cases also sufficient, because the flow can be described just up to the point of injection and combustion, and will therefore give a good understanding of the conditions before ignition. When working with highly 3 dimensional flows, troubles using hot-wires can occur because they cannot detect the direction of the flow. This can be helped by using more advanced 3D probes. Another disadvantage is that the hot-wire has to be calibrated, which can be very time consuming.
In this study, HWA was chosen as the best compromise because of the relatively simple test setup and frequency response.

2.1.1 Hardware for hot-wire anemometry

There are two methods for running a hot-wire, constant temperature anemometry (CTA) and constant current anemometry (CCA).
Running a hot-wire with the CCA method, the current sent through the wire is kept constant. This means, that the wire temperature is changing as function of the flow velocity and fluid properties. When the flow velocity is increasing, the cooling effect gets higher, decreasing the wire temperature and thereby the wire resistance. From Ohm's law, it can be seen, that when the resistance is decreasing the voltage is also decreasing, when the current is constant.

$$E = I \cdot R \tag{2}$$

Where E is the voltage, I is the current and R is the resistance. This drop in voltage, is the output from the hot-wire anemometer, and what can be measured and processed. In order to have a precise voltage output, a Wheatstone bridge is used, with the hot-wire integrated in the circuit. This can be seen in Figure 2, [2][fig. 4.1].

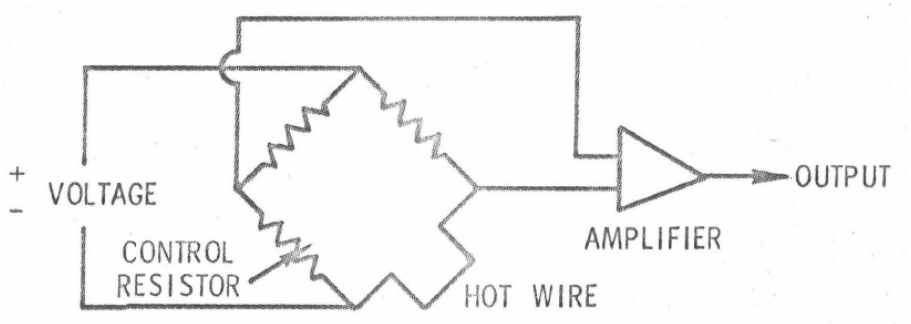


Figure 2: Example of Wheatstone bridge for CCA

Here it can be seen, that the hot-wire is put as one of the resisters in the Wheatstone bridge. The resistance in the three other resisters in the bridge are know, and thereby giving a relation between the wire resistance and the voltage output. In the circuit shown, an amplifier has been implemented in order to amplify the changes in the signal, gaining

a more precise measure of the voltage.
The CCA method is simple, but has a drawback due to thermal inertia of the hot-wire. This means that it takes time for the hot-wire to change its temperature. This is not a problem in simple flows, but in turbulence flows, the fluctuations in the flow lead to changes in the velocity. If the changes happen too fast, the hot-wire does not have time to find an equilibrium and will therefore lag behind the changes in the flow. This leads to incorrect results.

In order to encounter the problem of thermal inertia, the CTA method can be used. Here the temperature of the wire is kept constants via a feedback system, and thereby no movement of the wire temperature occurs. The feedback system is integrated in the Wheatstone bridge, as seen in Figure 3, [2][fig. 4.2].

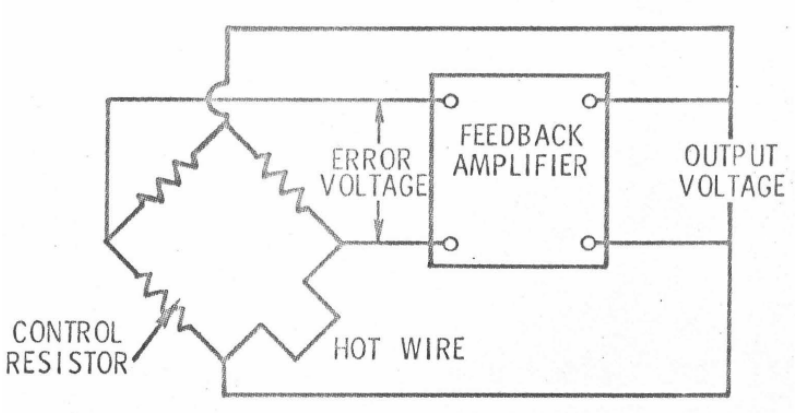


Figure 3: Example of Wheatstone bridge for CTA

The difference between the two circuits is that the input and output voltage is connected via the feedback system. When the fluid is cooling the hot-wire, the temperature and the resistance in the hot-wire will decrease, coursing an imbalance in the Wheatstone bridge, changing the output voltage as for the CCA system. The feedback system detects the drop in output voltage and will turn the input voltage up, thereby bringing the wire temperature back to its original temperature. The resistance is also brought up, bringing balance to the bridge. Because the feedback system is wired into the circuit, the temperature is corrected rapidly and thereby preventing the system for suffering from thermal inertia, and thereby increasing the frequency response of the system.
The wire temperature is determined from the overheat ratio, a, which is a ratio between the wire resistance at operating temperature and ambient temperature. From the overheat ratio, the operating temperature can be found as:

$$a = \frac{R_w - R_0}{R_0} \tag{3}$$

$$T_w = \frac{a}{\alpha} + T_0 \tag{4}$$

When making measurements in air, the overheat ratio is typically between 0.5 and 2 depending of the type of wire and application. The overheat ratio is regulated by adjusting the resistance in the Wheatstone bridge.

2.1.2 Governing equations

The wire temperature is an important parameter in the heat transfer equations. When talking about heat transfer, three types of heat transfer exist; convection, conduction and radiation. The radiation is in this case negligible and will not be taken into account. Conduction occurs from the hot-wire to the prongs, but are only 2-5% of the total heat transfer. Experimentalists have dealt with the problem by either neglecting conduction or using hot-wires with a wire length-to-diameter ratio bigger than 2000 in order to minimize the temperature gradient along the wire, which conduction to the prongs will create. Davis and Fisher have conducted an explicit solution for the mean wire temperature where conduction and convection are taken into account [12][equ. 10]. When using a direct calibration method, which will be described later, the conduction to the prongs will be taken into account by the calibration, and no further compensation has to be made.
The biggest part of the total heat transfer is the convection from the hot-wire to the flow. A relation between the flow velocity and voltage is needed in order to determine the velocity from the voltage output. The governing equation for the convectional heat transfer is described by:

$$\dot{q} = hA_w(T_w - T_0) \tag{5}$$

Where $\dot{q}$ is the convective heat transfer flux, h is the convection heat transfer coefficient and A_w is the surface area of the wire. The area is calculated as:

$$A_w = \pi D_w L_w \tag{6}$$

Where D_w is the diameter of the wire and L_w is the length of the wire. h is found from the Nusselt number, Nu, which is defined as:

$$Nu = \frac{hl}{k_f} \Leftrightarrow h = \frac{Nu\, k_f}{l} \tag{7}$$

Where k_f is the thermal conductivity of the fluid and l is the characteristic length, in this case the diameter of the wire. By combining Equation (5), Equation (6) and Equation (7) and substituting l with D_w the following equation is obtained:

$$\dot{q} = \frac{Nu\, k_f}{D_w}\pi D_w L_w(T_w - T_0) = Nu\, k_f \pi L_w(T_w - T_0) \tag{8}$$

The energy generated in the wire comes from the electric input, and is defined as:

$$\dot{q} = I^2 R_w \tag{9}$$

This is introduced in Equation (8) and by using Ohm's law the current through the wire can be found from the out put voltage and the total resistance of the wire and bridge, R_{total}. The total resistance can be measured and the following expression can be obtained:

$$I^2 R_w = Nu\, k_f \pi L_w(T_w - T_0) \Leftrightarrow \tag{10}$$

$$\left(\frac{E}{R_{total}}\right)^2 R_w = Nu\, k_f \pi L_w(T_w - T_0) \Leftrightarrow \tag{11}$$

$$Nu = \frac{\left(\frac{E}{R_{total}}\right)^2 R_w}{k_f \pi L_w(T_w - T_0)} \tag{12}$$

From this, the Nusselt number can be found. A relation between the Nusselt number and the Reynolds number, Re, is normally established through an empirical found relation. In hot-wire studies, the relation used is King's law:

$$Nu = A + B \cdot Re^n \tag{13}$$

Where A, B and n are constants found through calibration. From the definition of the Reynold number, the flow velocity can be found:

$$Re = \frac{Vl}{\nu} \Leftrightarrow V = \frac{Re\,\nu}{l} \tag{14}$$

Where l is the characteristic length, in this case D_w and ν is the kinematic viscosity. In this way, a relation between the flow velocity and the voltage has been established.

2.1.3 Calibration

Different types of calibrations are used when dealing with hot-wires. There is calibration for the relation between voltage and flow velocity and directional calibration. The velocity calibration can be done in different ways depending if the measurements are in 1D or 3D, and depending on the nature of the flow.

Direct calibration

In the pipe test the flow is turbulent, but the temperature and pressure are constant, and thereby also the fluid parameters. Because the nature of the flow is fairly simple, a calibration method called the direct method can be used. The direct method is a method where the hot-wire is calibrated in a control flow with a known velocity. The known velocity is then directly connected to the output voltage of the wire. A number of calibration points are made, and a calibration curve is created via a polynomial fit. The points in the calibration curve are taken within the same velocity range as expected to find in the flow. When the hot-wire is operated, the output voltage is measured and the corresponding velocity is found via the calibration curve. By using the direct calibration method, the equations above and conduction is taken into account y the calibration, and does not have to be used directly.

Temperature correction

In the engine test, the nature of flow is more complex. The temperature and pressure is no longer constant and thereby the fluid parameters either. This means that the simple direct calibration method cannot be used. In order to take the fluctuating temperature and pressure into account, temperature correction must be used.
First the relation between Nu and Re has to be established. A calibration similar to the direct method is carried out to define the constants in King's law. The flow is again at room pressure and temperature. The calibration points in the voltage-velocity diagram are saved. From the voltage, the Nusselt number can be calculated from Equation (12) and the Reynolds number can be calculated from Equation (14), at each calibration point. By curve fitting, the constants A, B and n can be found, and the King's law have been established.
The experiments can be carried out, and the raw voltage output signal from the hot-wire

is saved. With that, Nu can be found from Equation (12). But in order to make a correction of the elevated temperature and pressure, the thermal conductivity of the fluid, k_f, is corrected by using the following method from [14][equ. 3].

$$k_{cor} = k_0 \left(\frac{T_g}{T_0}\right)^{0.8} \tag{15}$$

Where T_g is estimated from a simple simulation explained in Section 2.6. With the corrected thermal conductivity, the corrected Nusselt number can be found. Through King's law, with the constants found from the calibration, the Reynolds number can be found. The velocity can now be calculated from Equation (14), but the viscosity also has to be corrected. The viscosity used in the correction is the dynamic viscosity by using [14][equ. 4].

$$\mu_{cor} = \mu_0 \left(\frac{T_g}{T_0}\right)^{0.76} \tag{16}$$

The relation between the dynamic and the kinematic viscosity is:

$$\mu = \nu\rho \tag{17}$$

Where ρ is calculated from the the ideal gas law:

$$\rho = \frac{p}{TR_{spec}} \tag{18}$$

Where R_{spec} is the specific gas constant for dry air which is equal to $287.058\frac{J}{kg \cdot K}$. With the correction of μ, the corrected velocity signal is hereby conducted and can be used for further processing.

Directional calibration

Directional calibration is a calibration which determines the position of the hot-wire. For the pipe test a 1D hot-wire was used. Because the wire cannot detect the direction of the flow, the position of the wire in the flow is important in order to get the wanted velocity signal. The wire has to be perpendicular to the main flow direction, as seen in Figure 1. For a 1D hot-wire, the directional calibration can be carried out by putting the hot-wire in the flow, and rotate it until it obtains the highest output value. In this way, the hot-wire will be positioned perpendicular to the main flow.
For a 2D or 3D wire, the directional calibration is highly important in order to obtain good results. The manufacture can carry out the directional calibration and provide the yaw and pitch coefficients needed in order to place the probe correctly in the flow. The directional calibration normally has to be made only once for the wire, since it relates to the geometry of the wire which will not change. If the yaw and pitch coefficients are not known, a calibration has to be made. This can be done by placing the hot-wire in a fixture which can adjust the pitch, yaw and roll coefficients. The probe is then placed in a control flow and rotated within a domain where the correct position can be expected to be found. The output voltage is recorded at each angular increment. The coefficients can then be found by comparing the output voltage to the angle.

2.2 Statistics

In order to describe a turbulent flow, statistics are often used. When working with turbulence the assumption of stationary flow cannot be applied, due to the nature of the turbulence. Instead the assumption of a statistical stationary flow can be applied. Meaning that two samples in time, but in same position, will not be the same. But if two series of samples are taking in the point, parameters as the average velocity and variance for each series will be the same, under the assumption of statically converged series. Thereby statistical stationary flow. Chosen statistical parameters will now be presented [14][equ: 5-7].
The mean velocity of a turbulent flow is found by:

$$\bar{V} = \frac{\sum_{i=1}^{N} V_i}{N} \tag{19}$$

Where $\bar{V}$ is the mean velocity, V is the raw velocity signal and N is the number of samples in the signal. In fluid mechanics The flows velocity is often divided in its mean velocity part and its fluctuation part. The mean velocity can be subtracted from the raw velocity signal in order to get the fluctuations of the flow:

$$V_f = V - \bar{V} \tag{20}$$

Where V_f is the fluctuations. An illustration can be seen in Figure 4

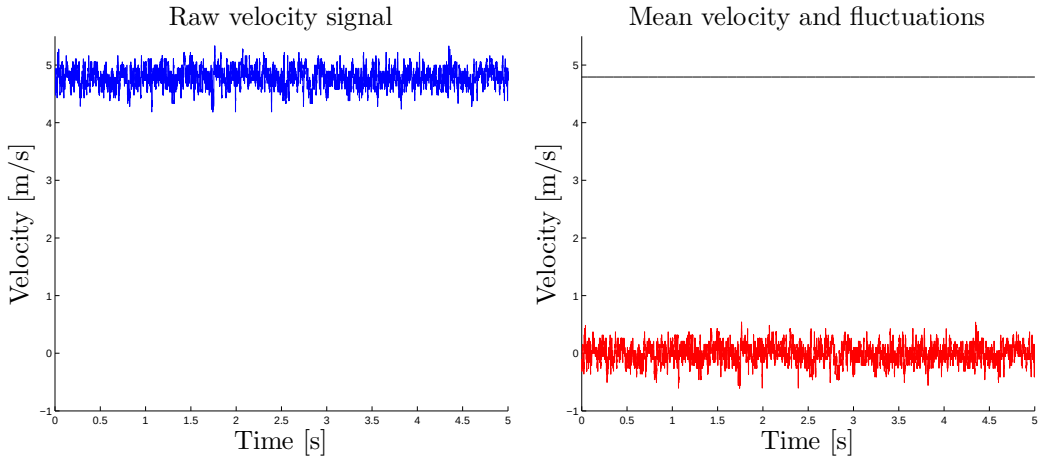


Figure 4: Illustration of division of mean velocity and fluctuations

The standard deviation of signal can be used to describe the turbulent intensity of the flow. The standard deviation , V_{std}, is calculated as:

$$V_{std} = \sqrt{\frac{\sum_{i=1}^{N} \left(V_i - \bar{V}\right)^2}{N}} \tag{21}$$

From the standard deviation, the variance, V_{var}. can be found as:

$$V_{var} = V_{std}^2 \tag{22}$$

Another parameter is the relative turbulent intensity, which describes how turbulent the flow is by comparing the turbulent intensity to the mean flow:

$$RTI = \frac{V_{std}}{\bar{V}} \tag{23}$$

2.3 Length and time scales

When characterizing a turbulent flow, the different structures in the flow can be described. The structures are eddies described by their size and life time. The eddies can be divided into three ranges, the energy containing range, the inertial subrange and the dissipation subrange.

In the energy containing range the largest eddies are found. These eddies are connected to the mean flow and are generated by pressure gradients and instabilities in the flow. The scales connected to this range are the integral time scale, T_{int}, and integral length scale, L_{int}. These describe the average life time and size, respectively, of the largest eddies that can exist in a flow before they are broken down to smaller eddies.

When the eddies are broken down, they enter the inertial subrange. This range is dominated by inviscid non-linear mechanisms, which means that the energy,brought into the inertial subrange by the largest eddies remains constant. If viscus effects had been present, it would have acted as an energy sink and would have taken energy out of the eddies. In this range the eddies are decreasing in size, so a fixed length scale cannot be found. Because of the non-viscous environment, the eddies are not dying out, so a fixed time scale cannot be found. The inertial subrange only exist for flows with high Reynolds number. If the Reynolds number is too low, the inertial subrange cannot be found and a transition from the energy containing rang directly to the dissipation subrange will occur.

In the dissipation subrange, the viscus effect are dominating. The viscus effects cause the eddies to die out and dissipate into heat. The scales connected to this range are the Kolmogorov time scale, T_{kol}, which describes the average life time of the eddies before they dissipate into heat. And the Kolmogorov length scale, L_{kol}, which describes the length of the average smallest eddies that can exist in a flow before they dissipate into heat.

These three ranges are part of what normally is called the energy cascade. The energy cascade describes how energy is transferred from external forces in the flow and into heat. The energy cascade is related to the energy budget of a turbulent flow, which can be described as [10][equ. 12.46]

$$\underbrace{\frac{\delta \bar{E}}{\delta t} + U_j \frac{\delta \bar{E}}{\delta x_j}}_{1} = \underbrace{\frac{\delta}{\delta x_j}\left(-\frac{U_j P}{\rho_0} + 2\nu U_i \bar{S}_{ij} - u_i \bar{u}_j U_i\right)}_{2} \underbrace{- 2\nu \bar{S}_{ij}\bar{S}_{ij}}_{3} + \underbrace{u_i \bar{u}_j \frac{\delta U_i}{\delta x_j}}_{4} \underbrace{- \frac{g}{\rho_0}\bar{\rho} U_3}_{5} \tag{24}$$

This equation can be divided into 5 terms. Term 1 is the time derivative of the change in mean flow kinetic energy. Term 5 is the loss to potential energy, which is not taken into account here. Term 2 is called the transport term. It can be shown, that this term does not affect the total rate of change of $\bar{E}$, but only transport the energy between regions.

This is in relation to the inertial subrange, where the energy is transported from bigger to smaller eddies. Term 3 is the viscous dissipation term. This is the term which is related to the dissipation subrange, where the kinetic energy is dissipated into heat. Term 4 describes the loss from mean flow kinetic energy to turbulent kinetic energy. This term represents the energy containing range, and describes the transition from mean flow to turbulent structures.

Another length scale is the Taylor microscale, λ. This length scale is not as physically related to the flow as the integral and Kolmogorov scales, but is a measure of two things. When looking at the fluctuations from Figure 4, the Taylor microscale is a measure of the average spacing between zero crossings. If the spacing is small, the eddies in the flow are also smaller. The Taylor microscale is also a intermediate scale which can be found between the integral and Kolmogorov scales.

Other types of scales can also be used to describe a turbulent flow, for example velocity scales and other kind of length and time scales. In this study, the scales described above will be used to characterize the flow.

2.4 Digital signal processing

The statistics are used to get a feeling of the flow, but if a more detailed analysis is needed, digital signal processing has to be used.
In this section a description of how to avoid contamination of the signal, design of measurements and how to calculate length- and time-scales and energy spectra are presented.

2.4.1 Aliasing and windowing

Two important phenomenons which have to be taken into account are aliasing and windowing. Both of these can contaminate the signal and thereby lead to incorrect results.

Aliasing

Aliasing is a phenomenon which occur when the sampling rate is too low. If the sampling rate is too low the samples recorded will not represent the real signal. An example can be seen in Figure 5, [15][p. 39].

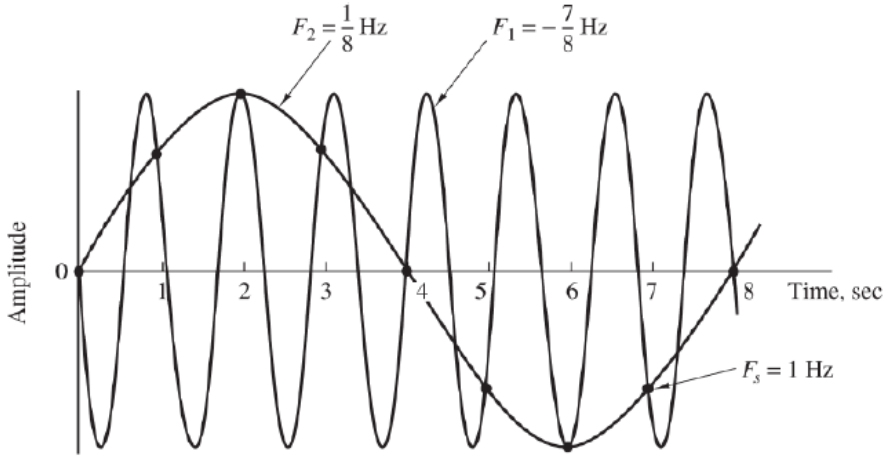


Figure 5: Sketch of aliasing

From the figure, two sinus curves can be seen. The curve with the frequency, F_1, represents a real signal. The dots on this curve represents recorded samples. The dots are the only knowledge about the signal, so when making a sinus curve between the points, the second sinus curve is constructed instead. Because the sample rate was too low to fully represent the real signal, an non-existing sinus curve has been constructed. This will lead to a transfer of energy in a energy-frequency plot from high frequencies to the lower ones, also called folding. An example of folding can be seen in fig. 6, [5][fig. 12.21].

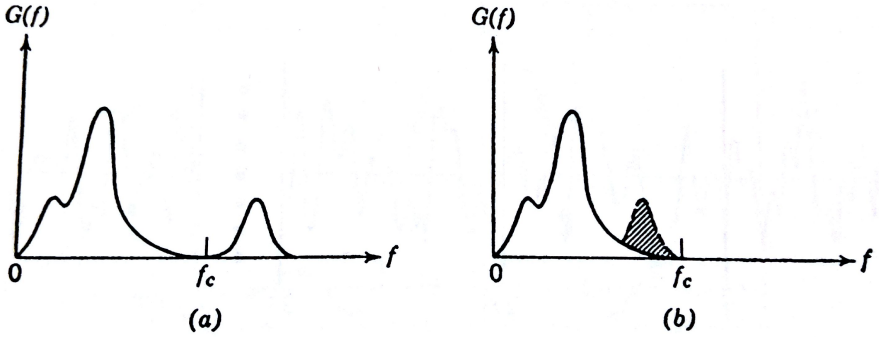


Figure 6: Folding due to aliasing

Here it can be seen how energy above a critical frequency, f_c, is folded back to lower frequencies, and thereby contaminates the distribution of energy.
To prevent aliasing, a low-pass filter can be applied with at cut-off frequency, $f_{cut-off}$. This has to be done before sampling, if the signal is contaminated it cannot be restored. There are two options for applying the low-pass filter. One is to know the maximum frequency, f_{max}, of the flow and apply a low-pass filter according to that. This can for example be done by experiments. When f_{max} is known, one has to take into account the Nyquist sampling criteria. Nyquist says that the maximum frequency that will not be aliased is half the given sampling rate. Meaning that in order to represent the real signal up to a frequency equal to f_{max}, one has to sample with a frequency twice as high. The

low-pass filter is applied to this frequency in order to prevent any further folding of energy, which can appear from noise. This can be described as:

$$f_{cut-off} = 2 \cdot f_{max} \tag{25}$$

A sketch of the method can be seen in Figure 7.

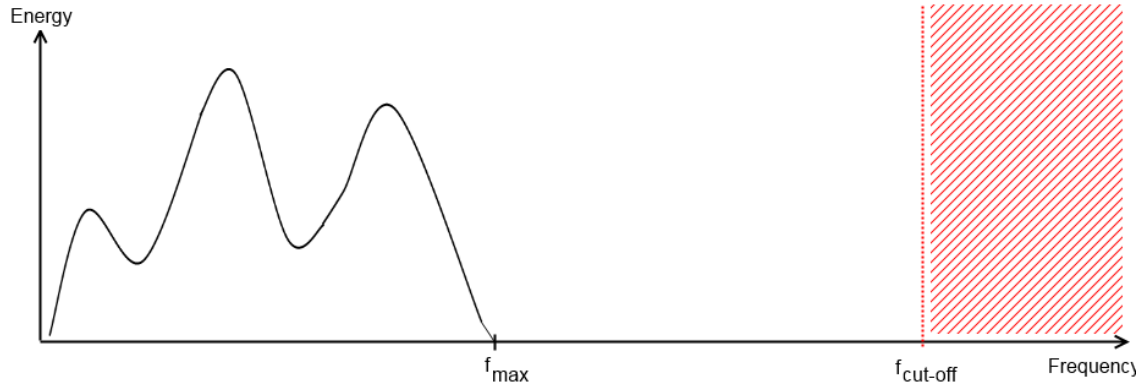


Figure 7: Sketch of method of highest frequency

An other way to avoid aliasing has to be used, when the equipment becomes the limiting factor. Some simple equipment has for example a few fixed settings for the low-pass filter. If the highest low-pass filter is not high enough to include the highest frequencies in the flow, the low-pass filter is applied and the sample rate is then set according to the filter. A sketch of this method can be seen in Figure 8.

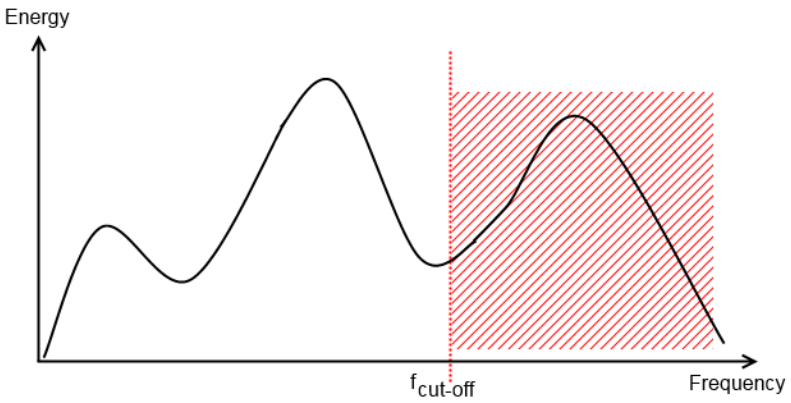


Figure 8: Sketch of filtering method

From the figure, the cut-off frequency of the low-pass filter is shown.It can be seen that all frequencies above the cut-off frequency is omitted. This can lead to a lack of data from the highest frequencies, but the signal will not be contaminated.

2.4.2 Sampling rate

The sample rate, SR, can be determined on background of the cut-off frequency found in Section 2.4.1, in order to prevent aliasing. In the first method the sample rate has to be

at least the same as the cut-off frequency, preferable higher, in order to fully represent the signal.

$$SR \geq f_{cut-off} \tag{26}$$

In the other method, where the equipment is the limiting factor, to fully represent the remaining part of the spectrum, one has to sample with at least double the frequency of the cut-off frequency according to Nyquist. This will lead to a sample rate of:

$$SR \geq 2 \cdot f_{cut-off} \tag{27}$$

In both cases, a low-pass filter has been applied. Because the filter used is not an ideal filter, the roll off time of the filter has to be taken into account. This is done by setting the filter to 3-5 times the Nyquist frequency. This mean for the second method that the sampling rate should not only be twice as big as the cut-off frequency, but 3-5 times bigger instead.

For other purposes, the sampling rate can be used for making uncorrelated measurements, and thereby deliberately not representing the signal. For measurements where only the data for velocity is needed and not for further processing, the sampling rate can be set so low, that the points obtained are uncorrelated, and will thereby give individual samples. The sampling rate can be found by:

$$SR_{uncorrelated} \leq \frac{1}{2T_{int}} \tag{28}$$

In this way, the samples are set so that two samples are not influenced by the same eddy.

Windowing

Windowing is a phenomenon which can occur when the measuring time is too short. The flow is seen as an infinitely long velocity signal, and only a small window of the signal can be captured when measuring. The width of the window is the measuring time. If the measuring time is shorter than the biggest integral time scale, then the biggest eddies will not be taken into account
Another effect of applying a window is that oscillations will occur when the signal is used afterwards in a fasts Fourier transformation, FFT. A sketch of this can be seen in Figure 9, [16].

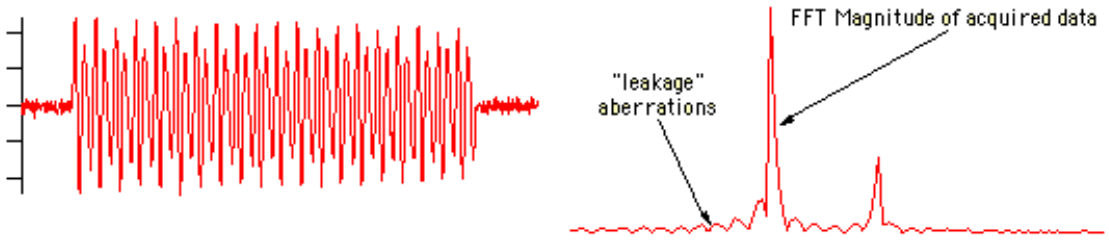


Figure 9: Effect of square window

From the figure a sample of a signal can be seen. When the sample is used in a FFT, the sudden change in the ends of the sample will lead to a leakage of energy in the frequency spectrum. In the figure this is seen as the small waves in the spectrum. These waves distributes energy from smaller frequencies in the energy spectrum to higher frequencies, resulting in a more flat spectrum.
In order to avoid cutting off the biggest eddies and reduce the leakage, the length of the sample has to be long. As a rule of thumb, the sample length should be 400 times longer than the biggest eddies in the system.

2.4.3 Block averaging

To obtain a readable signal from experimental data, it is necessary to perform block averaging. This subject is closely related to windowing. The obtained signal can fluctuate a lot, and thereby be hard to read. In order to obtain readable results without loosing information in the fluctuations, multiple series of data can be made, and averaged after having been processed. These series are called blocks. The length and amount of blocks needed depend on the flow and on the level of variability chosen.
Taking a series of samples in order to create a block is equivalent to apply a window on the continuous velocity signal. The block length can thereby be determined as 400 times the integral time scale, as mentioned before. The integral time scale can either be estimated mathematically or experimentally. A very rough estimation of the integral time scale in a pipe flow can be made from the diameter and center line velocity, which in this experiment can be estimated to:

$$T_{int,estimation} = \frac{D}{V_{cl}} = \frac{0.2m}{4\frac{m}{s}} = 0.05s$$

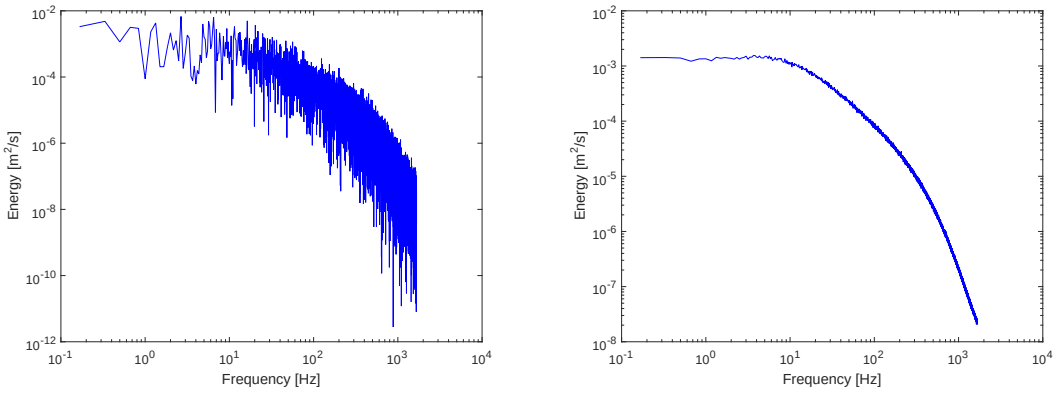
In this study, the integral time scale was estimated through testing. When the integral time scale has been estimated, the length can be found as:

$$t_{block} = T_{int,estimation} \cdot 400 = 20s \tag{29}$$

The amount of blocks depends on the level of variability, and can be found from:

$$N_{block} = \frac{1}{\varepsilon^2} \tag{30}$$

Where ε is the variability level. In this study a variability of 5% was used which is leading to 400 blocks.
Each block is processed individually, and the results of each block are afterwards averaged. An example of the effect of block averaging can be seen in Figure 10.



(a) (b)

Figure 10: Example of block averaging, where (a) is without block averaging and (b) is with block averaging

It can be seen from the figure, that the signal becomes more defined and thereby better for further processing.

From the sample rate, block length and block time, the total recording time and samples can be calculated by:

$$t_{total} = t_{block} \cdot N_{block} \tag{31}$$

and

$$N_{total} = t_{total} \cdot SR \tag{32}$$

2.4.4 Taylor microscale and integral time and length scale from autocorrelation function

A tool to investigate the correlation between two points within a signal is the autocorrelation function. The autocorrelation function takes a point of reference of the signal in time, and finds how dependent it is, to another point taken after a certain time step, τ. By finding the correlation between points in the signal, repeatable patterns in the signal can be found. When used in a turbulence context, the repeatable patterns are eddies moving over the hot wire. This means that the time before the signal repeats itself can be found, and thereby the time it takes the biggest eddies to get broken down to smaller eddies can be found. The autocorrelation function is defined as:

$$r(\tau) = \overline{V_f(t) \cdot V_f(t+\tau)} \tag{33}$$

This expression can be normalized by dividing with the variance, V_{var}, so the normalized autocorrelation function becomes:

$$R(\tau) = \frac{\overline{V_f(t) \cdot V_f(t+\tau)}}{V_{var}} \tag{34}$$

An example of a autocorrelation function can be seen in Figure 11.

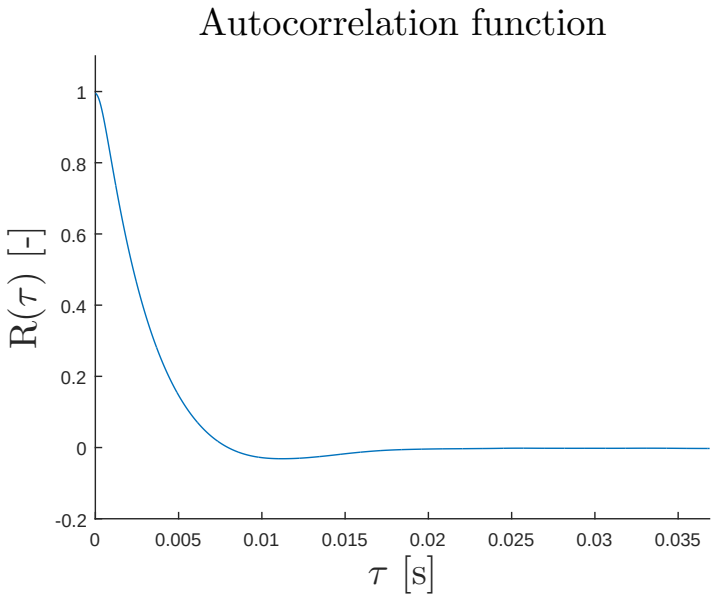


Figure 11: Sketch of an autocorrelation function

The figure shows, that when the two points compared are the same, there is a 100% correlation. As the time step gets bigger, the correlation between two points in time decreases.

From the autocorrelation function, the integral length scale and the Taylor microscale can be found. A graphical representation of this can be seen in Figure 12, [6][Fig. 6.10]. Here the integral time scale is denoted $\mathscr{F}$ and the Taylor microscale is denoted λ.

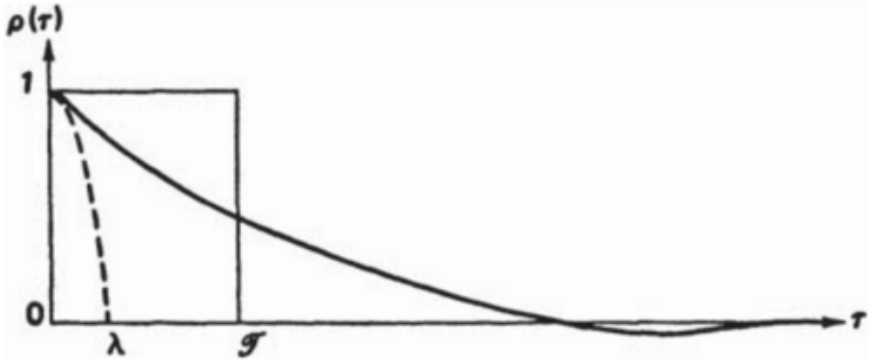


Figure 12: Graphical representation of integral time scale and Taylor microscale

The integral time scale is found by integrating the area under the autocorrelation function, and can be presented as shown in the figure. With this method, a proper result can be hard to obtain from experimental results. Therefore, different method to determine the integral time scale have been developed by P. L. O'Niell, D. Nicolaides, D. Honnery and J. Soria [11].

1. The most correct method is to integrate over the entire autocorrelation function.

$$T_{int} = \int_0^\infty R(\tau)\, d\tau \tag{35}$$

2. For an autocorrelation function with a negative region, integrate up to the minimum value.

3. Integrate up to the first zero crossing.

4. Integrate up to the value where the autocorrelation function drops below $\frac{1}{e}$

In the article it is concluded that by using method 1, the integral time scale will continue to grow when using more and more data, because the autocorrelation function does not stabilize at zero, when using real life data. Method 2 cannot be used if the autocorrelation function has not got a negative region. Method 4 can discard a lot of the function and thereby underestimate the integral time scale. The article concludes that the most precise result is obtained by using method 3, which was therefore used in this study. Looking at Figure 12, using method 3 will give the integral time scale as the are under the curve from $\tau = 0$ and until the curve crosses the x-axis. This area has been redistributed into a square with high 1 and width equal to the integral time scale.
When the integral time scale is found, the integral length scale can be calculated from:

$$L_{int} = T_{int} \cdot \bar{V} \tag{36}$$

The Taylor microscale can be found, as seen in Figure 12, as the parabola fitted to the autocorrelation function at origin. The Taylor microscale can also be found analytically, but it still has its origin in the autocorrelation function,. The curvature of the parabola can be approximated close to origin as [6][p. 211]:

$$\frac{d^2R}{d\tau^2}\,|_{\tau=0} \equiv \frac{-2}{\lambda^2} \tag{37}$$

By a Taylor expansion, an equation for the parabola can be obtained:

$$R(\tau) = \frac{-1}{\lambda^2}\tau^2 + 1 \tag{38}$$

By assuming a statistically stationary flow, it can be found that:

$$0 = \frac{d^2 V_{var}}{dt^2} \tag{39}$$

By combining Equation (37) and Equation (39) an analytical expression for the Taylor microscale can be obtained:

$$\lambda = \bar{V}\sqrt{\frac{V_{var}}{\overline{\frac{du}{dt}^2}}} \tag{40}$$

From this equation, the Taylor microscale can be found, and by using Equation (38) the parabola can be obtained.

Both when using the analytical and the curve-fitting method, noise in the signal has

to be taken into account, as it leads to a lowering of the autocorrelation function. Looking close to origin in Figure 11 it can be seen, that the function, despite the normalisation, is not exactly equal to 1. A close up of the curve can be seen in Figure 13.

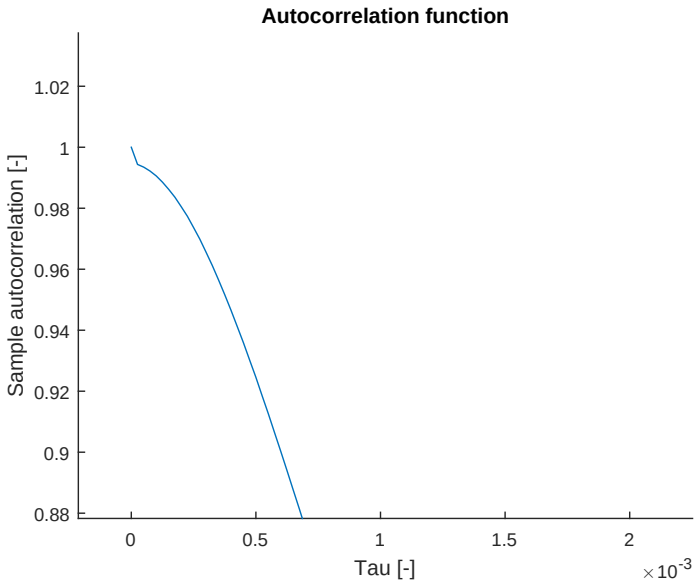


Figure 13: Close up of Autocorrelating function

Here it can be seen, that there is a drop from the point at x = 0 to the autocorrelation function itself. The influence from the noise can be explained from Equation (33). When noise, n, is taken into account the equation can be written as:

$$\overline{(V_f(t) + n(t)) \cdot (V_f(t+\tau) + n(t+\tau))} = \\ \overline{V_f(t) \cdot V_f(t+\tau)} + \overline{V_f(t) \cdot n(t+\tau)} + \overline{n(t) \cdot V_f(t+\tau)} + \overline{n(t) \cdot n(t+\tau)}$$

Assuming that the noise is white noise, n is totally random. This means that a correlation between the signal V_f and a random signal cannot be found, so the two mixed terms in the expression will become zero. A correlation between the random signal and itself cannot be found either, except when $\tau = 0$. Then there is a complete correlation and the autocorrelation function will at x = 0 be equal to 1, everywhere else this term will become zero. This leads to the peak in Figure 13. In order to compensate for this, the first point of the autocorrelation function was removed. A parabolic fit were then made for points 2 to 7. The fit were only made for some few points, because the fit is only valid close to the origin. Using points 2 to 7 gave the best results. From the parabola the point at x = 0 was found and put as a replacement for the point removed in the autocorrelation function.

2.4.5 Energy from signal

In the same manner as the autocorrelation function, the idea behind an energy spectrum is to break down the signal, and enable analysis of the individual eddies. In this way, the energy distributed over the energy cascade described in Section 2.3 can be analysed.

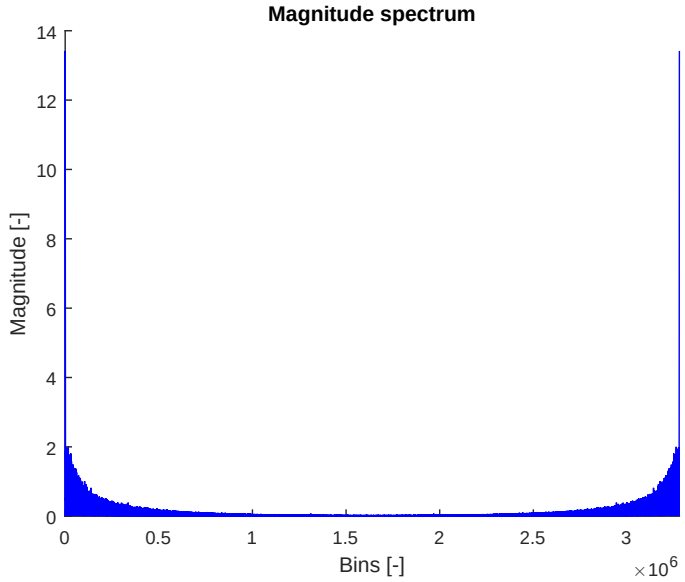


Figure 14: Example of magnitude spectrum

The velocity signal gained from measurements consist of many different sinusoids, which give a signal difficult to interpret when combined together. The sinusoids can be isolated by there frequency via a Fast Fourier Transformation (FFT):

$$\hat{V} = \int_{-\infty}^{\infty} e^{-i2\pi ft} V_f \, dt \tag{41}$$

Where $\hat{V}$ is the output of the FFT - the velocity fluctuations in the frequency domain. In an experimental case, the discrete FFT must be used. Meaning that the input to the FFT is not a continuous signal, but consists of many individual points. The integration is changed to a sum with the limits 0 to $n-1$, where n is the number of points in the discrete signal. The FFT was carried out using MatLab's function *fft*. The function in MatLab is not taking into account the dt from Equation (41), so the dt is found as the time between two samples, $dt = \frac{1}{SR}$, and multiplied with the *fft* function.
From the FFT, a complex signal is obtained. By finding the modulus of the complex signal, the magnitude of the signal can be obtained, $|\hat{V}|$. The length of the output signal from the FFT is equal to the length of the input signal. The magnitude of the signal versus the number of samples, bins, can be plotted as seen in Figure 14. The energy density can be found from the magnitude:

$$E = \frac{|\hat{V}| \cdot |\hat{V}^*|}{T} \tag{42}$$

T is the sample time and $|\hat{V}^*|$ is the magnitude of the complex conjugated signal. A sketch of a point from the complex signal, with its complex conjugated, can be seen in Figure 15.

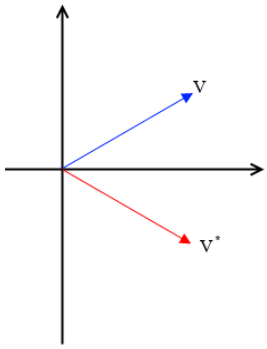


Figure 15: Sketch of modulus of a point and its complex conjugated

Here it can be seen, that taking the modulus of the vectors will give the same length. This means, that Equation (42) can be rewritten into:

$$E = \frac{|\hat{V}|^2}{T} \tag{43}$$

The bins can be converted into frequencies by making a vector which starts from 0 and runs up to a value equal to the SR, and is n elements long. By using the energy density and the frequency vector an energy spectrum can be obtained:

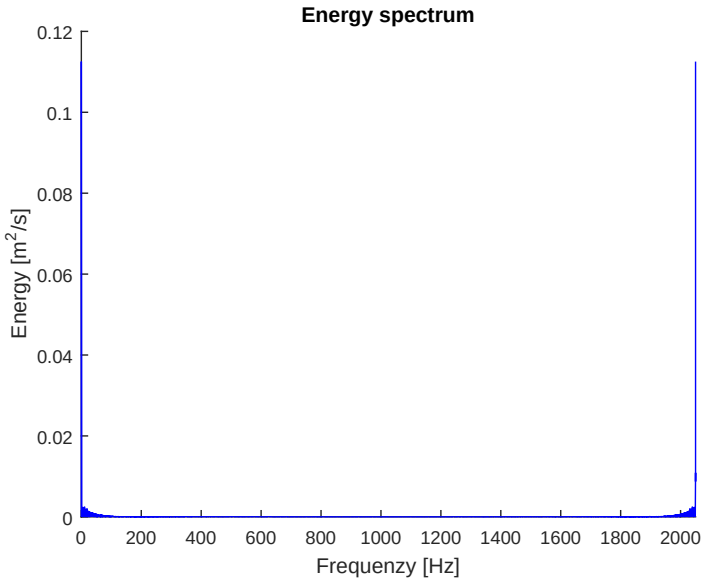


Figure 16: Example of energy spectrum

It can be seen, that the spectrum is symmetrical around $\frac{SR}{2}$. It is common practise to only look at one side of the spectrum. This is called the one sided energy spectrum. The one sided energy spectrum can be plotted in a log-log plot. An example can be seen in Figure 17.

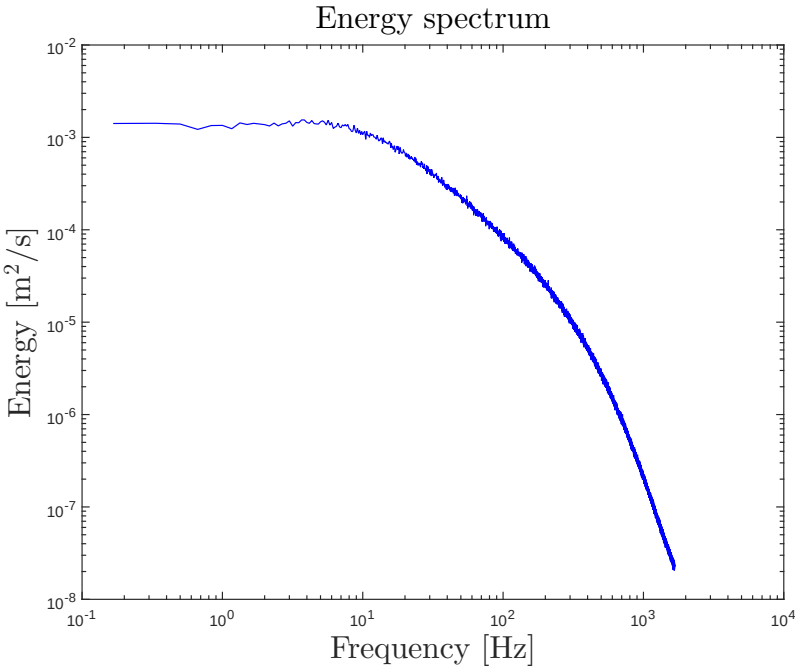


Figure 17: Example of energy spectrum

The energy spectrum represents the distribution of energy at different eddy sizes. The energy in the energy spectrum is the turbulent kinetic energy of the eddies.

It could be expected, that the energy spectrum would have a maximum where a special eddy size would contain the most energy. This is also often the case, an example can be seen in Figure 18, [8][figur 4.12]. This energy spectrum has been normalized with its maximum value and plotted as function of wavenumber. Here it is clearly seen, that the spectrum is having a maximum. The difference between Figure 18 and Figure 17 is that Figure 18 has been obtained with a 3D system whereas Figure 17 has been obtained by a 1D system. When using a 1D system, the wire cannot detect the direction of the flow, and will thereby see flow velocities from other direction as contributions to the main flow direction. This means that energy from all three directions are added up, and thereby flatten out the start of the spectrum. A 3D system can resolve the energy into the different direction, and the curve will thereby have a maximum.

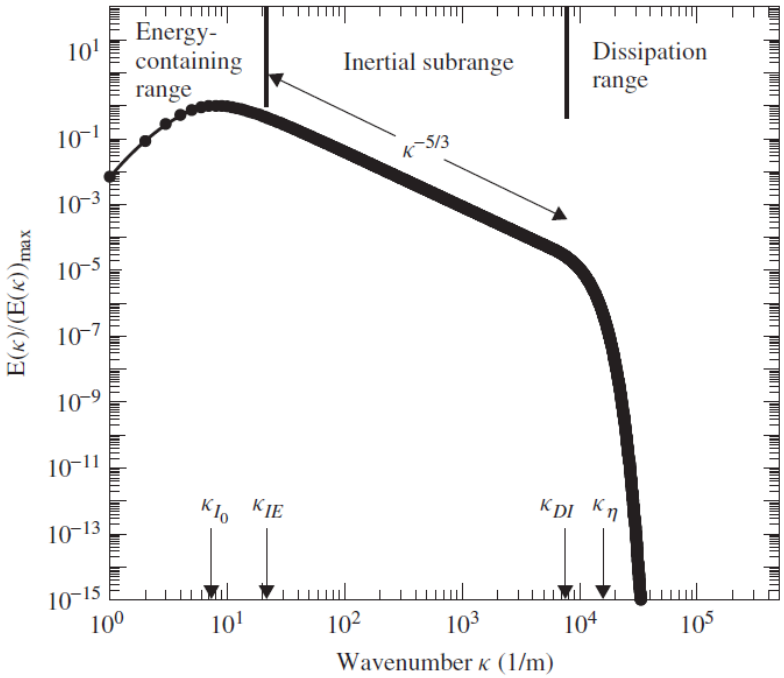


Figure 18: Example of energy spectrum with maximum

The energy spectrum has a relation to V_{var}. The variance can be found by integrating the probability density function over all frequencies. The variance found from Equation (22) and the one from the integration can be compared and thereby make part of a validation of the energy spectrum. Because this is done for only the on sided energy spectrum, a factor of 2 has to be multiplied, making the integration looking like:

$$V_{var} = 2 \cdot \int_0^{\frac{SR}{2}} E\, df \tag{44}$$

Because the curve consists of discrete points, the area under the curve has to be estimated. This can be done by different methods. It was chosen to use the MatLab function *trapez*, which estimates the area via the trapezoidal method.

An other way of calculating the energy spectrum has been proposed by Kolmogorov, which uses the correlation [8][equ. 4.213].

$$E(\kappa) = C\varepsilon^{\frac{2}{3}}\kappa^{\frac{-5}{3}} \tag{45}$$

Where C is a constant, often put to 1.5, ε is the dissipation rate and κ is the wavenumber which is inversely proportional to eddy size. In the inertial subrange, the dissipation rate can be left out, meaning that a line only depending on the $\frac{-5}{3}$ factor is obtained. This can also be used in an energy spectrum which uses frequencies instead. It leads to the rule that the energy spectrum has to follow a $\frac{-5}{3}$ slope in the inertial subrange. An example of this slope can be seen in Figure 19.

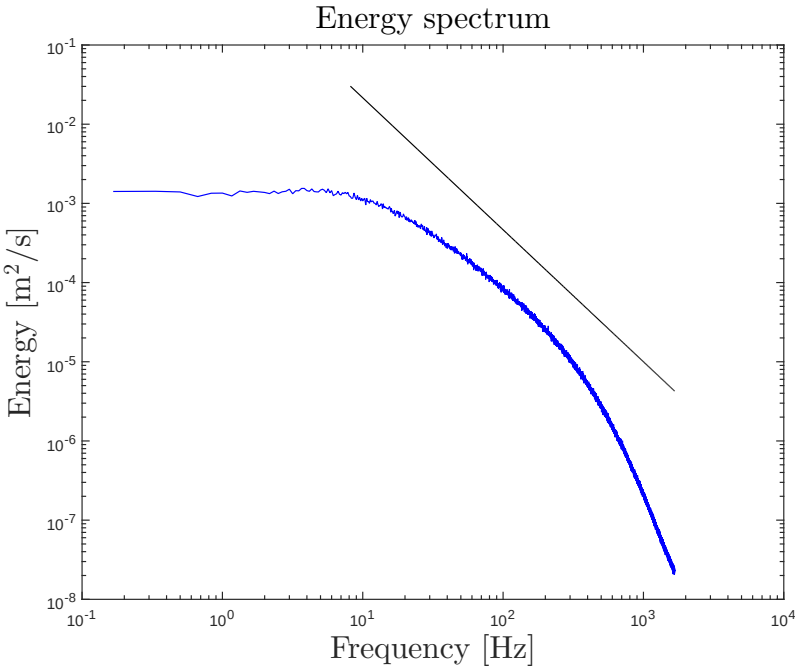


Figure 19: Kolmogorov's $\frac{-5}{3}$ rule

Here it can be seen how the energy spectrum follows the $\frac{-5}{3}$ slope at the area of the inertial subrange.

Another way of obtaining the energy spectrum is from the autocorrelation function. The energy spectrum can be obtained by making a FFT with the non-normalized autocorrelation function, r, as input, and the autocorrelation function can be obtained by making an inverse FFT (IFFT) with the energy spectrum as input. This also means, that the integral time scale can be obtained by the energy spectrum [5][p. 444]. The conversion from the autocorrelation function to the energy spectrum can be written as.

$$E(f) = \int_{-\infty}^{\infty} e^{-i2\pi ft}\, r(\tau)\, d\tau \tag{46}$$

The limits $[-\infty; \infty]$ refer to the two-sided energy spectrum and can thereby be rewritten as in Equation (44). Furthermore, from isolating $r(\tau)$ Equation (34) the following expression can be obtained.

$$\begin{aligned} E(f) &= 2 \cdot \int_{0}^{\infty} e^{-i2\pi ft}\, R(\tau) \cdot V_{var}\, d\tau \\ &= 2 \cdot V_{var} \cdot \int_{0}^{\infty} e^{-i2\pi ft}\, R(\tau)\, d\tau \end{aligned} \tag{47}$$

Looking at Equation (35) it can be seen, that an expression containing the integral time scale can be obtained by setting $f = 0$.

$$
\begin{aligned}
E(0) &= 2 \cdot V_{var} \cdot \int_0^\infty e^{-i2\pi 0t} \, R(\tau) \, d\tau \\
&= 2 \cdot V_{var} \cdot \int_0^\infty R(\tau) \, d\tau \\
&= 2 \cdot V_{var} \cdot T_{int} \Leftrightarrow \\
T_{int} &= \frac{E(0)}{2 \cdot V_{var}}
\end{aligned} \tag{48}
$$

$E(0)$ can be found by extrapolating the energy spectrum to $f = 0$. This is an other method to validate the energy spectrum.
The Taylor microscale and Kolmogorov length scale can be used to validate the energy spectrum. In Figure 18 two length scales are shown, κ_{DI} and κ_η. These indicate the borderline between the inertial subrange, and the dissipation subrange and the Kolmogorov length scale respectively. The Taylor microscale can be expected to be found near κ_{DI}, determining the transition from the inertial subrange to the dissipation subrange. When entering the dissipation subrange, the Kolmogorov length scale can be expected to be found close after the kink in the energy spectrum.

2.5 Kolmogorov time and length scale

The Kolmogorov time and length scales can be found directly from the velocity signal and thereby be compared to the energy spectrum to verify the results. The Kolmogorov scales can be found from the dissipation rate, ε, [5][equ. 2.79].

$$
\varepsilon = 15\nu \frac{V_{var}}{\lambda^2} \tag{49}
$$

Where λ is the Taylor microscale found in Equation (40). The factor of 15 is a result of the simplification of the dissipation rate when the assumption of isotropic turbulence has been applied. The Kolmogorov time and length scales can be found as a correlation of ν and ε as:

$$
T_{kol} = \left(\frac{\nu}{\varepsilon} \right)^{\frac{1}{2}} \tag{50}
$$

$$
L_{kol} = \left(\frac{\nu^3}{\varepsilon} \right)^{\frac{1}{4}} \tag{51}
$$

In order to compare the length scales (integral, Kolmogorov and Taylor) to the energy spectrum, the lengths have to be converted to frequencies, which can be done by:

$$
f_{int} = \frac{\bar{V}}{2\pi L_{int}} \tag{52}
$$

$$
f_{kol} = \frac{\bar{V}}{2\pi L_{kol}} \tag{53}
$$

$$
f_\lambda = \frac{\bar{V}}{2\pi \lambda} \tag{54}
$$

With this, the length scales can be plotted together with the energy spectrum.

2.6 Simulation of engine pressure and temperature

In order to perform the temperature correction, the temperature of the cylinder gas at any time has to be known. The gas temperature can be estimated by a simple simulation. The simulation is performed in the period of the engine cycle where the valves are closed and a confined volume is obtained. This is in the period 180 CAD to 540 CAD. It can be assumed that the cylinder during the closed period is isolated. This means that no gas leaks or heat losses occurs. This lead to the assumption of isentropic compression and expansion.

$$pV^{\gamma} = Cont. \tag{55}$$

Where γ is the specific heat ratio which can be assumed constant and equal to 1.4. The volume of the cylinder can be calculated from engine data as function of CAD, [13]. By using the engine data the displacement volume, V_d, can be found from.

$$V_d = \frac{\pi}{4} B^2 S \tag{56}$$

From the displace volume and the compression ratio the compression volume, V_c can be found from.

$$V_c = \frac{V_c}{\epsilon - 1} \tag{57}$$

With the compression volume, the volume as function of CAD, $V(\theta)$ can be found from a geometrical expression.

$$V(\theta) = V_c + \frac{\pi B^2}{4} \left(R\left(1 - \cos(\theta)\right) + \frac{R^2}{2L} \sin(\theta)^2 \right) \tag{58}$$

By assuming that the gas at 180 CAD is at atmospheric pressure and temperature, the pressure as a function of CAD can be found from the Equation (55).

$$p_1 V_1^{\gamma} = p_2 V_2^{\gamma} \Leftrightarrow p(\theta) = p_{atm} \left(\frac{V(\theta)|_{180CAD}}{V(\theta)} \right)^{\gamma} \tag{59}$$

The pressure curve obtained is an ideal process, unlike the process in the engine where pressure and heat losses are present. The engine pressure can be measured and the simulation can thereby be corrected. This is done with a correction factor which can be found from.

$$CF = \frac{p_{max,measured}}{p_{max,simulaiton}} \tag{60}$$

The correction factor has to be found from every wanted engine speed, because the maximum pressure is changing as function of engine speed. With the correction factor, the pressure curve can be corrected. With the corrected pressure known at every CAD, the temperature as function of CAD can be found from the ideal gas law.

$$T(\theta) = \frac{p(\theta) V(\theta) T_{atm}}{p_{atm} V(\theta)|_{180CAD}} \tag{61}$$

With the temperature as function of CAD, the temperature correction can be performed.

3 Pipe test set-up

The equipment used to carry out the pipe tests and the test procedure will here be presented. The purpose of the pipe test is to obtain a knowledge of the hot-wire anemometer method and detect possible difficulties regarding test set-up and execution of the test. The measurement will be carried out in one direction parallel to the main flow at two position downstream and throughout the radius of the pipe.

3.1 Test set-up

The pipe test was carried out by using a $6m$ long pipe with a diameter of $200mm$, see Figure 20. The end of the pipe is connected to a pipe-system which leads the air away from the set-up. The flow is driven by a fan placed after the pipe. In the pipe-system an orifice plate is mounted and a manometer is measuring the pressure drop over the orifice plate. This was used to adjust the speed of the fan.
In order to insert the hot-wire into the pipe, holes with a diameter of $10mm$ were drilled in the pipe and sealed with tape when not used, in order to fulfil the conservation of mass. The test set-up was made for another test, where gas mixing was investigated [4], so a hole for gas injection was already made. This hole indicates the zero-point in the longitudinal direction of the pipe. The measurements carried out in this pipe test were made 10 and -5 pipe diameters respectively, from this zero-point.

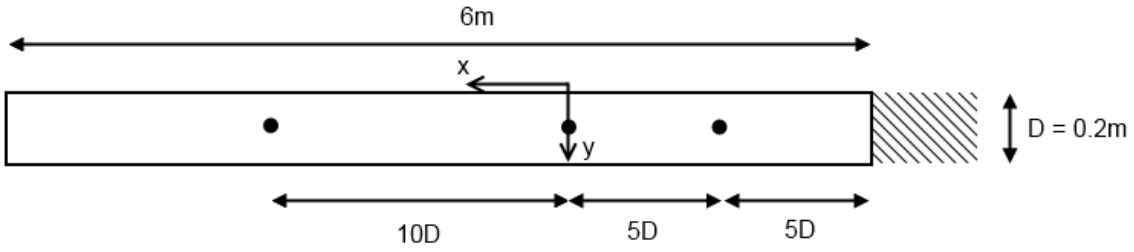


Figure 20: Sketch of pipe

In order to get a more stable flow into the pipe, a filter and a bank of small tubes was mounted at the inlet of the pipe. This can be seen as the hatched area in Figure 20. It was discovered, that without the filter and tubes, the flow could be affected by people walking close to the pipe or if doors in the laboratory were opened. With the filter and tubes, the flow was much less sensitive to the environment in the laboratory.

A fixture to hold and determine the radial position (y-position) of the hot-wire during testing was produced and can be seen in Figure 21.

Figure 21: Probe support fixture

In order to calibrate the hot-wire, a jet was used and located next to the pipe. The jet consists of a fan which blows air through a diffuser. The air flows through a filter and a bank of tubes, which makes the flow laminar. Hereafter, the air flows through a nozzle. The fan speed can be controlled and a micro manometer measures the pressure difference over the nozzle. By knowing the pressure difference and reading the ambient pressure and temperature from meters in the laboratory, Bernoulli's equation can be used and a relation between the pressure difference and flow velocity can be established.

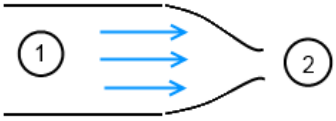


Figure 22: Sketch of jet

$$\begin{aligned}
\frac{p_1}{\rho} + \frac{1}{2}V_1^2 + gz_1 &= \frac{p_2}{\rho} + \frac{1}{2}V_2^2 + gz_2 \Leftrightarrow \\
\frac{p_1}{\rho} + \frac{1}{2}V_1^2 + 0 &= \frac{p_2}{\rho} + \frac{1}{2}V_2^2 + 0 \Leftrightarrow \\
p_1 - p_2 &= \rho\frac{1}{2}\left(V_2^2 - V_1^2\right)
\end{aligned}$$

By using conservation of volume flow rate the expression can be rewritten into

$$\begin{aligned}
v_1 A_1 = v_2 A_2 \Leftrightarrow v_1 = v_2\frac{A_2}{A_1} \Rightarrow \\
p_1 - p_2 = \rho\frac{1}{2}v_2^2\left(1 - \left(\frac{D_2}{D_1}\right)^4\right)
\end{aligned} \tag{62}$$

The nozzle has an inlet diameter of 0.24m and outlet diameter of 0.1m, which leads to.

$$p_1 - p_2 = \rho \cdot v_2^2 \cdot 0.485 \tag{63}$$

With this equation, an Excel sheet was made in order to go from pressure difference to velocity and back when the hot-wire was calibrated. An example of the Excel sheet can be seen in Appendix A.

3.2 Equipment

The equipment used in this test set-up can be described using a measuring chain. The measuring chain used can be seen in Figure 23 (Modified version of [7][Fig. 1]).

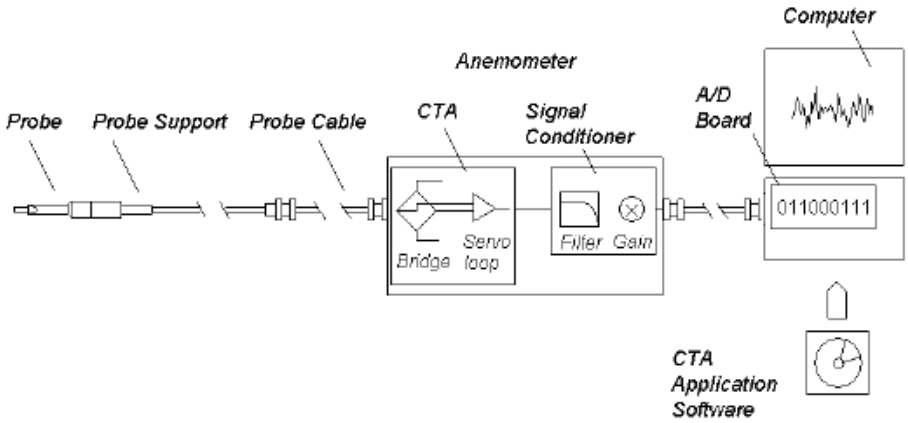


Figure 23: Measuring chain

The measuring chain shows the elements of the test set-up and how these are connected. The individual elements will now be described.

3.2.1 Hot-wire probe and wiring

The hot-wire used in this experiment is a 55P11 single sensor hot-wire probe from Dantec Dynamics [1]. The probe has a wire with a length and diameter of $1.25mm$ and $5\mu m$, and the wire material is platinum-plated tungsten.
The hot-wire probe is kept in place by a $235mm$ probe support. From the probe support a MiniCTA system is connected via a $4m$ probe cable. The length of the probe support and the probe cable are important to know, because the length will influence the overall electrical resistance of bridge in the CTA system.

3.2.2 MiniCTA system

The MiniCTA system used is a MiniCTA 54T30 from Dantec Dynamics. The MiniCTA can be divided into two parts, the CTA part and the signal conditioner part.
The CTA part is the Wheatstone bridge that keeps the temperature of the hot-wire constant, as described in Section 2.1. A diagram of the bridge can be seen in Figure 24 [7][p. 51].

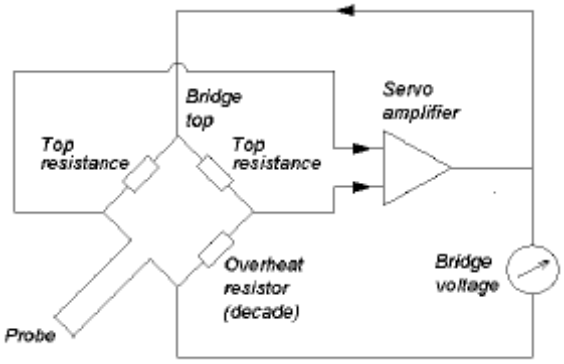


Figure 24: Diagram of CTA diagram

The signal conditioner part of the MiniCTA, is where the low-pass filter, overheat ratio, signal gain and offset can be chosen. The low-pass filter can be set to $1kHz$, $3kHz$, $10kHz$ or switched off. The switched off position is still limited and corresponds to a filter around $50kHz$. A signal gain, which amplifies the signal from the hot-wire, can be switched on or off. To help the signal to be within the limits of the MiniCTA's output voltage $(0-5V)$ an offset can be switched on or off. The MiniCTA and its options can be seen in Figure 25.

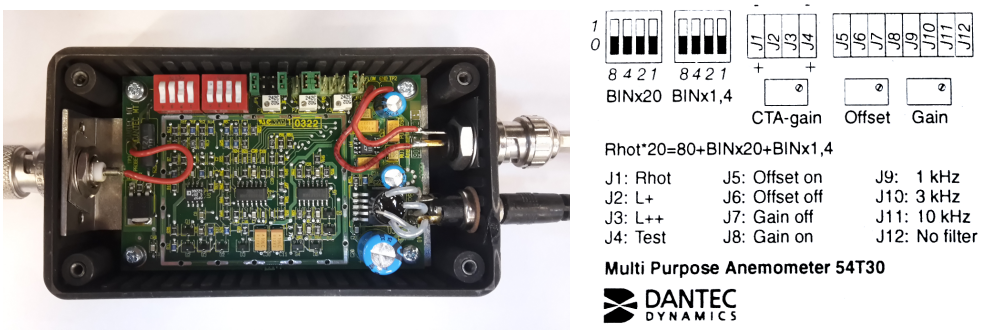


Figure 25: MiniCTA during measurements and its options

The options can be chosen by moving the green switches to their desired positions. The red and white buttons adjust the overheat ratio of the CTA system. Their positions are determined by the CTA-software when setting up the hardware in the program. For the test, the gain and offset were switched off and a low-pass filter of $10kHz$ was chosen.

3.2.3 A/D Board

To convert the analog signal from the miniCTA into a digital signal, a BNC-2120 A/D board from National Instruments was used. The probe cable was connected to the A/D board in channel 2 and with the option of measure floating signal source switched on. The A/D board has a 12-bit resolution which, according to DanTec [7][sec. 5.2.3], makes the CTA system able to run without any amplification of the CTA signal. The input ranges for the A/D board are $\pm 5V$ or $\pm 10V$. The output of the miniCTA is $0-5V$, so the A/D board was set the the $\pm 5V$ setting.

3.2.4 Software

The digital signal from the A/D board was collected by a computer with the DanTec software "StreamWare Basic" installed. When creating a job in the software, the measurement chain is entered in the program, which then calculates the overall resistance in the CTA system. When determining the overheat ratio, the software gives the setting of the red and white switches, as seen in Figure 25. Next, the sampling rate and number of samples are set in the software. A timer function can be used, which can repeat a measurement with a fixed time interval. The software has a calibration procedure for direct calibration, which creates a calibration curve. The calibration curve is used for the following measurements. The results obtained was a series of velocities with an associated time stamp. These results can be written to a txt. file, which was used as input to a MatLab script. The MatLab script processed the data and created plots.

3.3 Test procedure

The test procedure is here presented, and describes the order and method of how the set-up, calibration and measurement were carried out. Before the tests were performed, a series of pre-test was made. These were made in order to get a first feeling of the flow. From these a feeling of the eddy size and lifetime was obtained, and was used in order to design the real tests.
First the hardware set-up was made. The parts in the measuring chain were connected, without the power switched on. The measuring chain was entered in the software and the overheat ratio was set to 0.8. The miniCTA was set up by turning off the gain and off-set and the low-pass filter was set to $10kHz$. The overheat ratio was adjusted in the miniCTA from the instructions in the software. After the hardware was connected and adjusted, the probe was placed in the stream of the jet and the power for the miniCTA was switched on.
It was controlled that the probe was working via the 'Run online' function in the software, and the direct calibration was started. The probe was placed perpendicular to the flow, so it simulated the orientation, that it would be mounted in, in the pipe. Because the flow velocity in the pipe would be around $5\frac{m}{s}$, the calibration was carried out in the range from $2\frac{m}{s}$ to $10\frac{m}{s}$ with 10 calibration points. The procedure of the calibration is, that the software ask for the ambient temperature and pressure, and a flow velocity. The corresponding pressure difference over the jet nozzle was found via Equation (63) and the jet was adjusted via the micro manometer so it matched the required flow velocity. The software found the corresponding voltage output and made a calibration point in a velocity-voltage plot. The 10 calibration points were determined in this manner over the calibration range and the software fitted a 4th order polynomial to the points in order to create the calibration curve.
The fan for the pipe was switched on and set to match a pressure different over the orifice at $345Pa$, and was running for a while before the measurements started. The probe was, without disconnecting it, moved from the jet to the pipe and was mounted in the probe support fixture. Two different series of measurements were carried out, a series where the velocity profile was measured and a series where the turbulent structures were measured.

Velocity profile

For the velocity profile measurements, the sample rate and number of samples were set to $50Hz$ and 50 samples respectively, giving a recording time of $1s$. The sample rate was chosen from the lifetime of the biggest eddies, based on the pre-test and was set lower in order to insure independent samples, as described in Section 2.4.2. Measurements were taken at -5 diameter (-5D) in the x-direction and from 0.000 diameter to 0.900 diameter with steps of 0.025 diameter in the y-direction. It was decided to stop at 0.900 diameter, i order to prevent the hot-wire to hit the other side of the pipe, which could break the hot-wire. For each measurements, the ambient temperature was recorded and the temperature was within the range of $23.7°C$ and $23.9°C$. The ambient pressure was recorded before and after the measurements and was within the range of $101375Pa$ and $101461Pa$. From this it was concluded that the fluid properties could be assumed as constant. The velocity results were saved as an txt. file. The name of the file and other data were written in an Excel sheet which can be found in Appendix B.
The probe and fixture were moved to the 10 diameter (10D) position and the procedure was repeated. Here the temperature was within the same range as before and the pressure was within the range of 101461 Pa and 101470 Pa, so again the fluid properties were assumed to be constant.

Turbulent structures

For the turbulent structures measurements, the sample rate was determined from Equation (27) and was set to $40000Hz$ due to the imperfect filter. For these measurements, a variability of 5% was chosen, which means that 400 blocks were needed. The block length was determined from the pre-test and Equation (29) and was found to be just under $6s$. Round up to $6s$ and with a sample rate of $40000Hz$, this equals 240000 samples. The software was programmed to take 400 blocks of 240000 samples with $1s$ between each block in order to make the blocks independent.
After the velocity profile measurements at -5D, three turbulent structures measurements were made at 0.500D, 0.675D and 0.850D. From the pre-test it was seen, that the velocity profile was turned towards the pipe-wall on the opposite side of the hole for the hot-wire support, and the hole seams to disturb the flow. This will be explained in section Section 4.1. In order to prevent the measurement to be affected by this, the measurements was taken at the other side of the center of the pipe. After the probe was moved to the 10D position the turbulent structures measurements were performed in the same manner. The ambient temperature and pressure were recorded and the temperature was within a range of $23.0°C$ and $24.0°C$ and the pressure was within a range of $101150Pa$ and $101470Pa$. Again the fluid properties were assumed to be constant.

The data files from all the measurements were collected and were processed by MatLab. All the velocity profile measurements was processed at the same time using the MatLab script seen in Appendix C. The turbulent structures measurements produced data files which when processed made the computer run out of RAM. So the six series had to be processed individually. The results from each run in MatLab were saved in a database. The databases was then collected and run by another MatLab script, which collected the data and created the plots. The script for creating the data bases and the script for collecting the data bases can be seen in Appendix D.

4 Pipe test

The results from the pipe tests are here presented. The velocity profile measurements are presented, and hereafter the turbulent structures measurements.

4.1 Velocity profile

A mean flow velocity profile for the -5D and 10D positions have been obtained using the procedure in Section 3.3. The two velocity profiles can be seen in Figure 26.

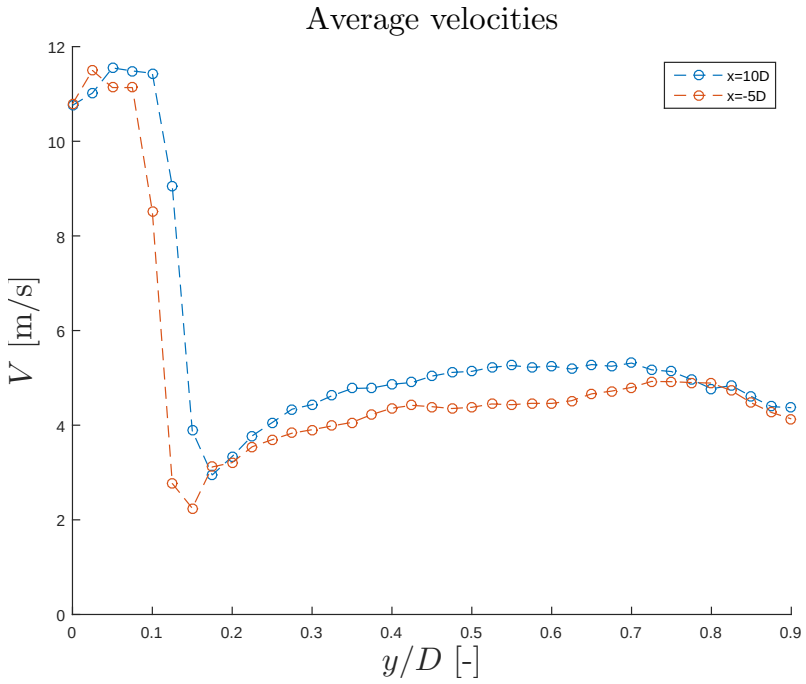


Figure 26: Average velocities as function of diameter

From the figure it can be seen, that there is a peak in the velocity close to 0D. This can be due to the hole for the hot-wire, which was drilled in the pipe. Air can be sucked through the hole, and thereby adds to the cooling effect of the hot-wire, resulting in a higher velocity. In order to see the velocity profiles more clearly, the fist part of the curve was disregarded. With the new boundaries the velocity profiles were more clear.

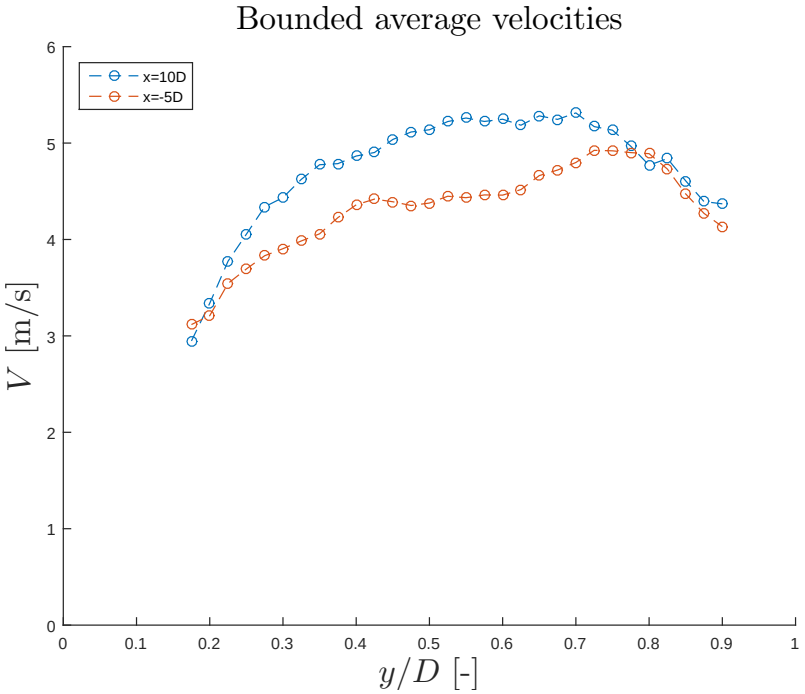


Figure 27: Average velocities as function of diameter with boundaries

It was expected to see velocity profiles with a parabolic shape and with the maximum value at the center line. Maybe a slightly more flat profile for the -5D position could be expected, because it is located closer to the tubes and filter. For the profile at 10D, a parabolic shape can be seen, but the maximum value is pushed to towards one side of the pipe. Looking at the profile at -5D, the profile is more flat, but is also pushed to the right and has a bump at around 0.7D - 0.8D. This behaviour can be explained by the entry of external air through the hole and the entry to the pipe. The air coming through the hole might push the center of the profile away from the center line. Also, if a perfect entry to the pipe was needed, a nozzle had to be fitted at the entry of the pipe. This would ensure a more uniform inflow and a more well defined boundary layer. These two factors can explain the shape of the profile.

The profiles were compared to similar results obtained by L. W. B. Browne and A. Dinkelacker [9][fig. 1]. The tendencies of the profiles are the same, both starting to drop at around 0.8 of the maximum velocity. The maximum point are lower because the curves are pushed towards the right.

The profile for the relative turbulent intensity has also be calculated. Because the RTI is a function of the mean velocity, Equation (23), the profile was expected to follow the averge velocity but had a parabola which is flipped compared to Figure 27. The first part of the curve was again removed and the profile for the RTI can be seen in Figure 29.

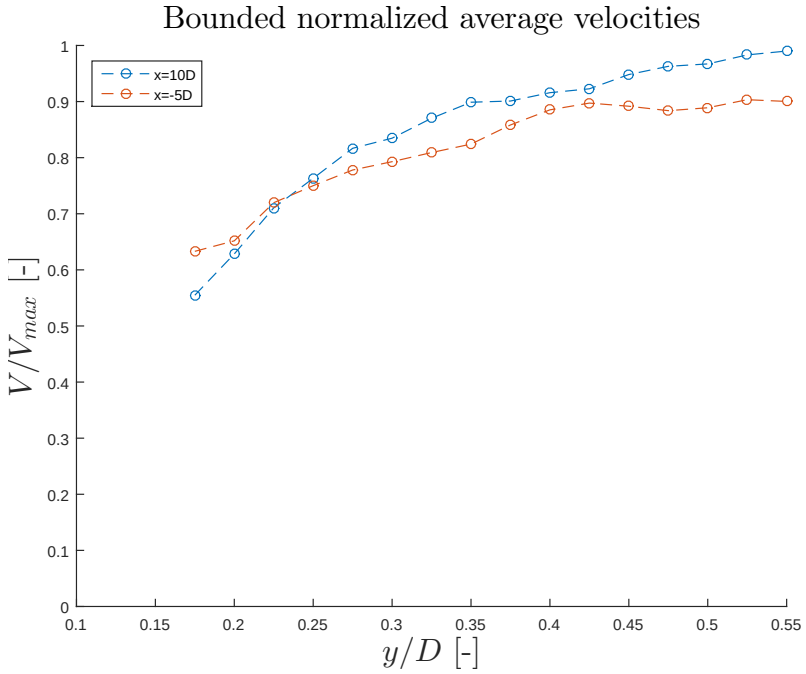


Figure 28: Normalized average velocities as function of diameter with boundaries

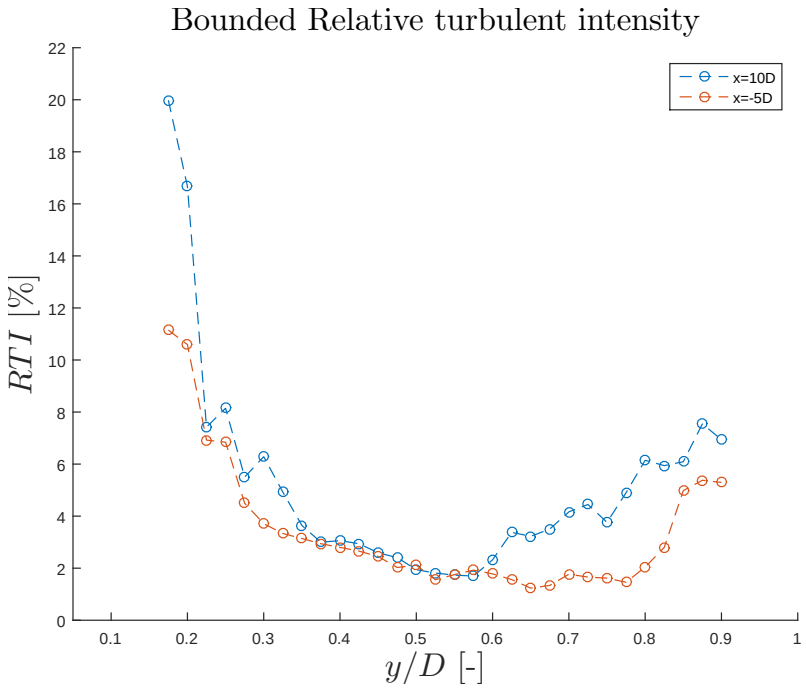


Figure 29: RTI as function of diameter with boundaries

The figure shows, that the profile follows the velocity profile as expected, and had the same tendencies. It can be seen, that the general level of RTI is under 8%, with a peak close to the hole. This indicates a flow where the mean flow is the dominating factor with a low turbulent intensity.

The RTI is in general lower at the -5D position compared to the 10D position. The velocity is also lower, indicating a smaller standard deviation. This could signal that the flow is more ordered at -5D, which make sense because it is closer to the tubes which forces the eddies to be more uniform. Whereas the more developed flow has a more broad variety of eddies.

4.2 Turbulent structures

The test for determining the size and lifetime of the eddies in the flow was conducted at -5D and 10D in the x-direction and in 0.500D, 0.675D and 0.850D y-direction, resulting in a total of six test. The results have been divided into three groups; average velocity and RTI comparison, autocorrelations and energy spectra.

4.2.1 Average velocity and relative turbulent intensity

The average velocities of the mean flow in the points were calculated and compared to the velocity profiles. The velocities can be seen in Figure 30.

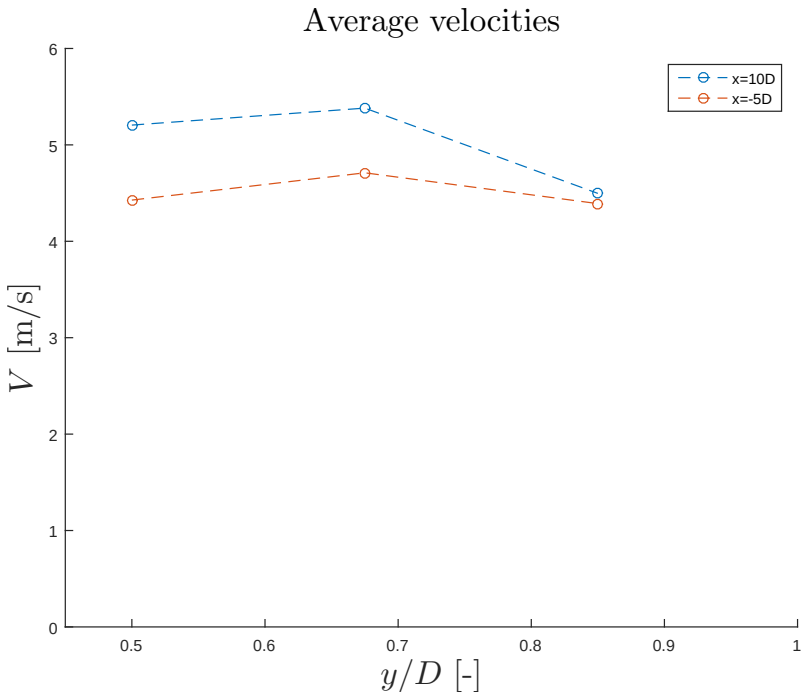


Figure 30: Average velocity

As expected, it can be seen, that the velocities agree with the ones obtained from the velocity profile. Also the RTI was calculated and compared. The results of the RTI can be seen in Figure 31.

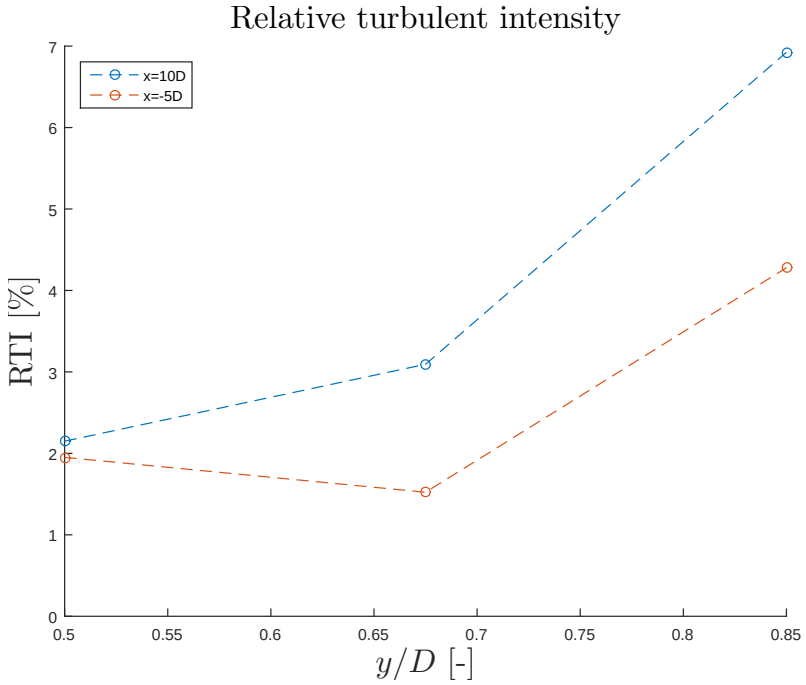


Figure 31: Relative turbulent intensity

The results for the measurements at 10D agree with results obtained from the RTI profile. The results for the measurements at -5D seems to be a lower. This can be due to the number of measurements point taken in the two measurements. In the RTI profile measurements averaging was done with 50 points and in these measurements, the averaging was done for 96 mio. points. In general the two tests agree on the tendencies in the flow.

4.2.2 Autocorrelations

In order to obtain the integral length and times scales, the autocorrelation function was calculated from Equation (34). The autocorrelation function for the six measuring points, can be seen in Figure 32.

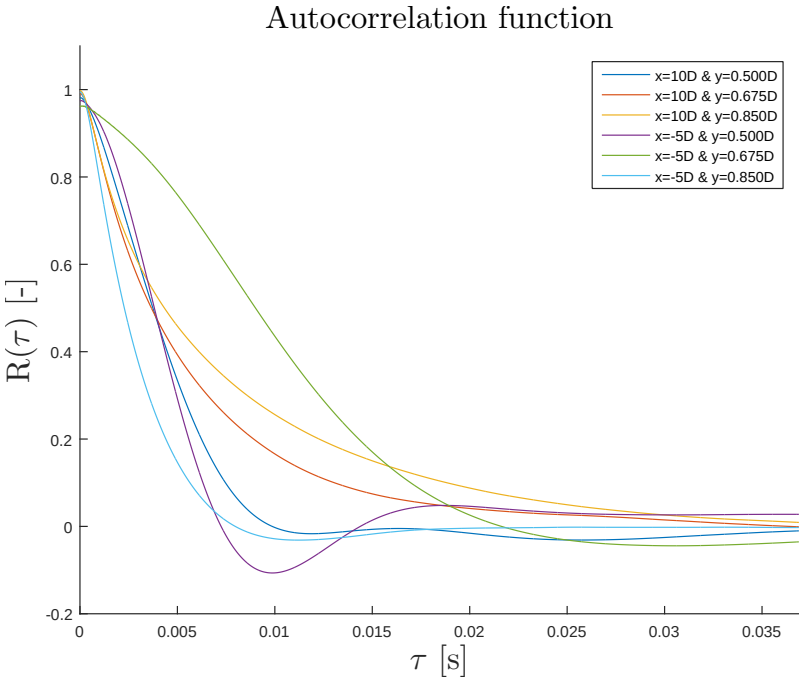


Figure 32: Autocorrelation

By using the method described in Section 2.4.4, the integral time scale could be calcualted by integrating the area under the autocorrelation function from $\tau = 0$ until the first zero crossing. By using Equation (36), the integral length scale can be calculated from the integral time scale. The results can be seen en Table 1.

x-position	**y-position**	$\boldsymbol{T_{int}[s]}$	$\boldsymbol{L_{int}[m]}$
10D	0.500D	0.0040	0.0208
10D	0.675D	0.0057	0.0309
10D	0.850D	0.0074	0.0332
-5D	0.500D	0.0038	0.0168
-5D	0.675D	0.0093	0.0439
-5D	0.850D	0.0027	0.0119

Table 1: Integral time and length scales

The integral time and length scales have been plotted in Figure 33.

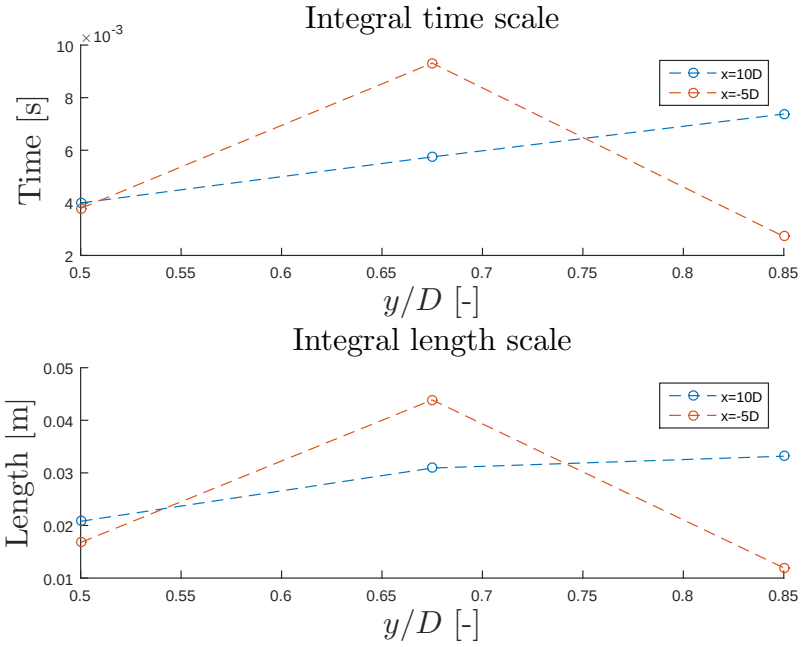


Figure 33: Integral time and length scales

Looking at the results for the 10D position, it can be seen that the integral time scale is growing when approaching the wall, and thereby also the integral length scale. This can be explained by the development of the turbulent boundary layer, which will grow thicker and when the flow is fully develop, will grow to cover the entire cross-sectional area. It seems like the flow is not fully developed at 10D, but the boundary thickness has grown to some extent, meaning that there is a region in the center of the pipe, which is not within the turbulent boundary layer. This can also be seen from the RTI profile, which shows that the region closes to the wall is having the highest turbulent intensity.
This increasing trend of the integral time scale is not the case for the results at the -5D position. The results for 0.500D and 0.850D are closer together, which can be expected because of the tubes and filter. The results at 0.675D seem to act differently. The auto-correlation function for this point also does not look like the others. Looking at Figure 26, the velocity in this area is high, so it seems like the flow inlet has an effect on the flow, which vanishes as the flow gets more developed.

The Taylor microscale was calculated using Equation (40). The results of the Taylor microscale can be seen in Table 2.

x-position	**y-position**	$\boldsymbol{\lambda}[m]$
10D	0.500D	0.0007
10D	0.675D	0.0010
10D	0.850D	0.0015
-5D	0.500D	0.0005
-5D	0.675D	0.0004
-5D	0.850D	0.0010

Table 2: Taylor microscales

The Taylor microscale have been plotted in Figure 34.

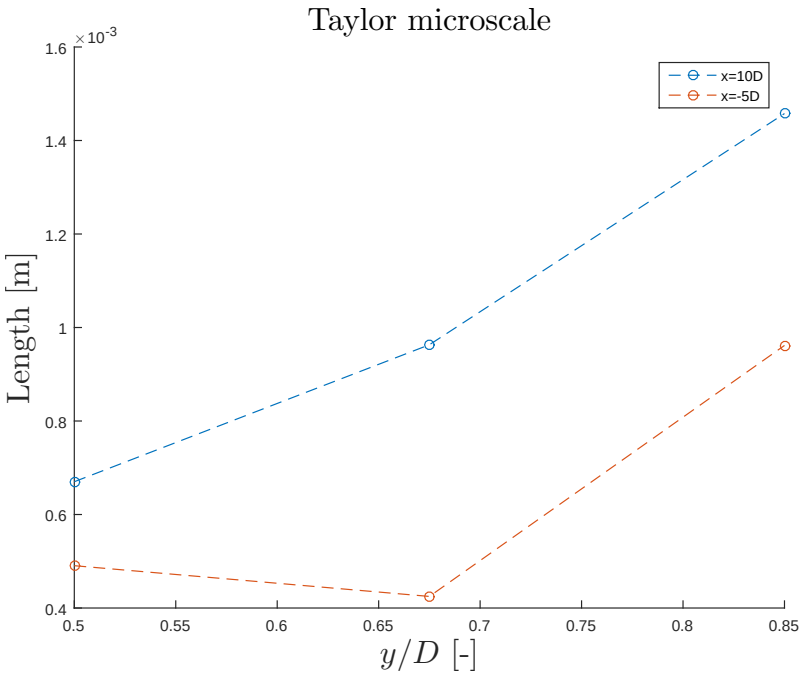


Figure 34: Taylor microscale

It can be seen, that for the 10D position, the Taylor microscale increases as it gets closer to the wall. This is the same tendency as seen from the integral time scale. For the -5D position, the 0.500D and 0.850D positions follows the same tendencies, but are in general lower than the one for the 10D position. This can be due to the tubes. The point for the -5D position at 0.675D has a Taylor microscale which is the same as the the one at 0.500D, despite having an integral time scale which is much higher. This indicates that the eddies decrease in size faster here, than at the other points.

The integral time scale and the Taylor microscale can be plotted together with the autocorrelation function, as seen in Figure 12. The integral time scale is equal to the area under the autocorrelation function, and this area can be collected into a square with hight equal to one and width equal to the integral time scale. In this manner the integral time scale can be plotted as a vertical line in the autocorrelation plot. The Taylor microscale can be plotted by using Equation (38). The integral time scale and the Taylor microscale can be seen in Figure 35.

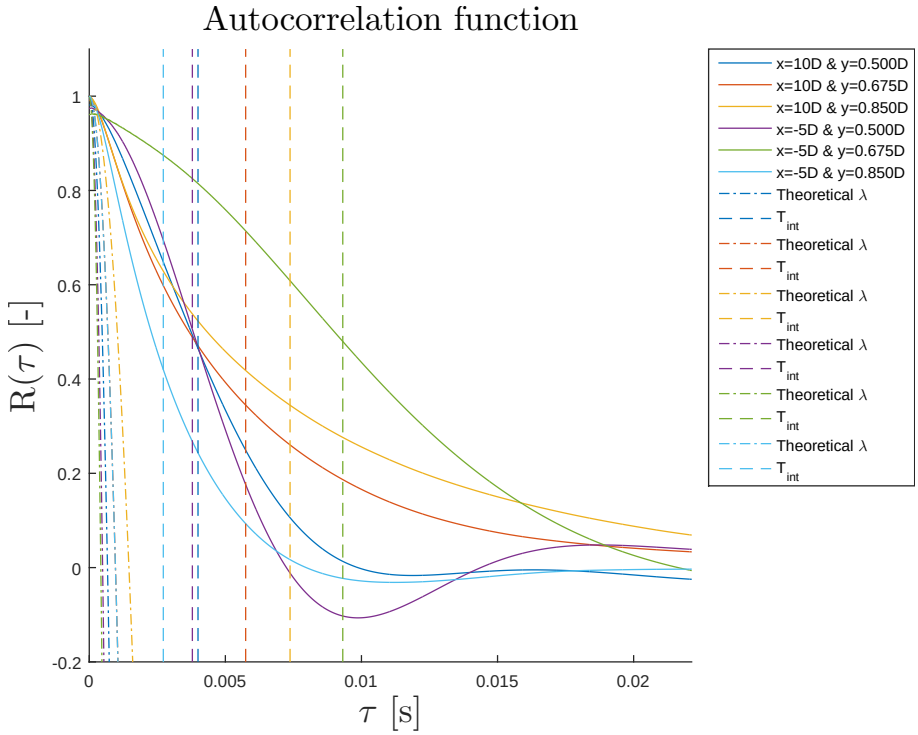


Figure 35: Autocorrelation with T_{int} and λ

From the figure it can be seen, that the lines of the integral time scales seem to match the area under the autocorrelation functions. And it can be seen, that the Taylor microscale can be read from its curve's zero-crossing with the x-axis. This indicates that the autocorrelations functions obtained seem to be correct.

4.2.3 Energy spectrum

The energy spectra from each of the six measurement points were conducted using the procedure explained in Section 2.4.5. By using MatLab's FFT function, which is equal to Equation (41), multiplied with $dt = \frac{1}{SR} = \frac{1}{40000} = 2.5 \cdot 10^{-5}s$ the output of the FFT is obtained. By using the FFT-output in Equation (43) with $T = t_{block} = 6s$ the signal energy density can be found. Transforming the bins into frequencies, the energy spectra can be plotted in a log-log plot, and can be seen in Figure 36.

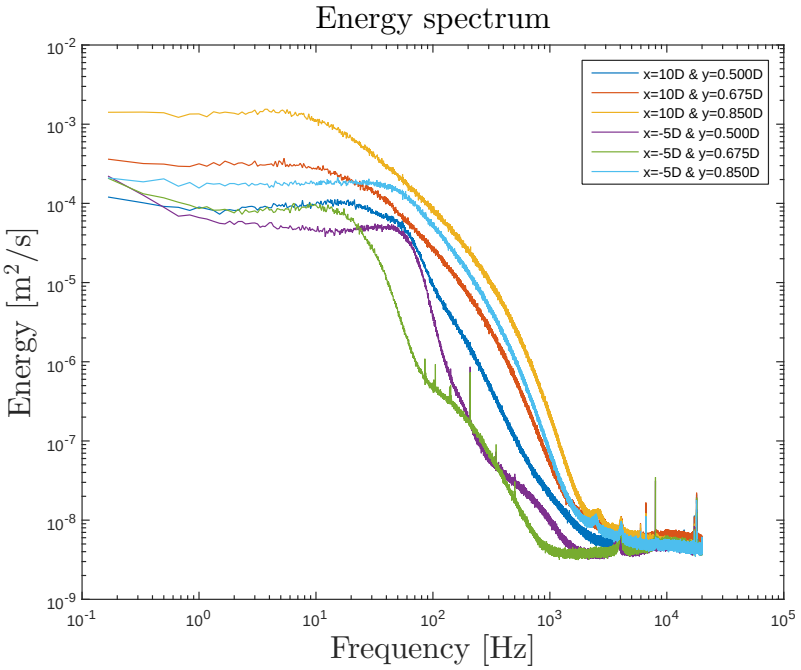


Figure 36: Energy spectrum

In order to validate the spectrum, the area under the curve was calculated and compared to the variance of the velocity signal, as seen from Equation (44). The results can be seen in Table 3.

x-position	**y-position**	V_{var}	**Area**
10D	0.500D	0.012527	0.012527
10D	0.675D	0.027737	0.027737
10D	0.850D	0.096977	0.096978
-5D	0.500D	0.007466	0.007466
-5D	0.675D	0.005150	0.005150
-5D	0.850D	0.035314	0.035314

Table 3: Comparison of area under the energy spectra and variance of velocity signal

Here it can be seen, that the area under the energy spectra and their respectively variance are almost identical. This indicates that the spectra have been conducted correctly. Because the autocorrelation function and the energy spectrum are related, the integral time scale can also be found from Equation (48). This was done, by fitting a curve to the first part of the curve. The coefficients of the fit were used to create a function, which could be evaluated at $f = 0$, and thereby find $E(0)$. The fit and results for the integral time scale can be seen in Figure 37.

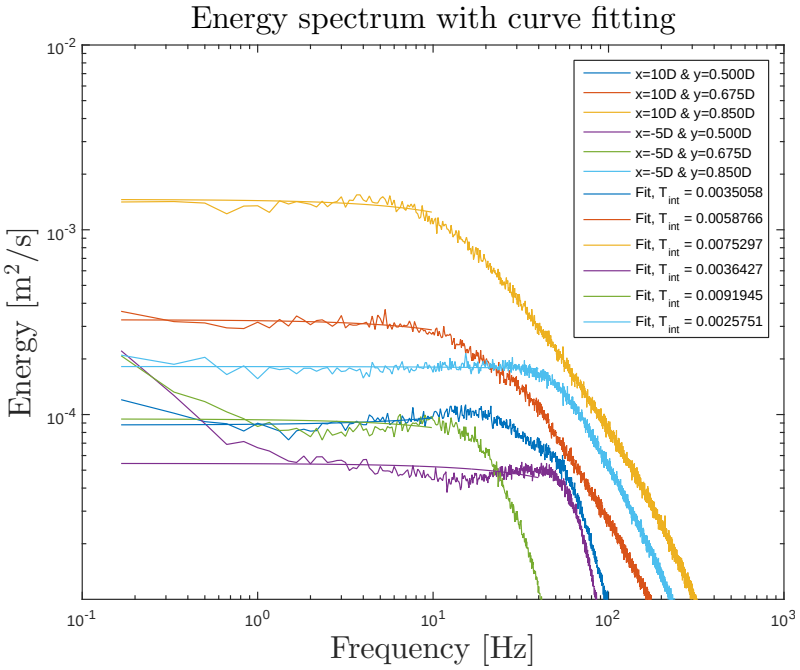


Figure 37: Energy spectrum with fits and estimated T_{int}

The results for the integral time scale from the curve fitting and from the autocorrelation function have been compared in Table 4.

x-position	**y-position**	T_{int} **autocorrelation** [s]	T_{int} **energy spectrum** [s]
10D	0.500D	0.0040	0.0035
10D	0.675D	0.0057	0.0059
10D	0.850D	0.0074	0.0075
-5D	0.500D	0.0038	0.0036
-5D	0.675D	0.0093	0.0092
-5D	0.850D	0.0027	0.0026

Table 4: Comparison of T_{int} from autocorrelation and energy spectrum

From the table it can be seen, that there is a good agreement of the integral time scales. The difference between the two methods seems to come from the curve fitting, so also from this point of view the energy spectra seem to be correct.

With the energy spectrum verified, an analysis of the curves can now be made. Looking at Figure 36, the three ranges of eddy sizes can be seen.
The first parts of the spectra are flat, and represent the energy containing range. As explained in Section 2.4.5, a flat first part of the curve indicates that the measurement have been obtained with a 1D probe, which is also the case. It can be seen, that the curves start out with different energy levels. Looking at the 10D position, the trend is that the energy level is higher closer to the wall than at the center line. This is also the trend for the -5D position, but not as pronounced as for the 10D position. The rising energy level can be explained by the relative turbulence intensity. The RTI is higher close to the wall

and thereby indicates a higher level of turbulence compared to the main flow. This means that most of the mean flow kinetic energy has been transformed into turbulent kinetic energy, which will lead to an increased energy level. Close to the wall, the RTI is higher for the 10D position, compared to the -5D position, which indicates that the mean flow is more dominating at the start of the pipe, and energy level will be lower as a result of that. For the -5D position, the two curves at 0.500D and 0.675D are having a more similar energy level, compared to the one at 0.850D. This can indicate, that the turbulent boundary layer at the start of the pipe, is thin, and only affects the curve closes to the wall. At the 10D position, the boundary layer seems to have grown, and the curves at this position have therefore a more different energy level.
The next step of the curve is at the first kink, which indicates the transition from the energy containing range to the inertial subrange. This is also where the integral length scale can be expected to be found. The integral length scales was transformed from lengths to frequencies via Equation (52) and plotted in Figure 64.

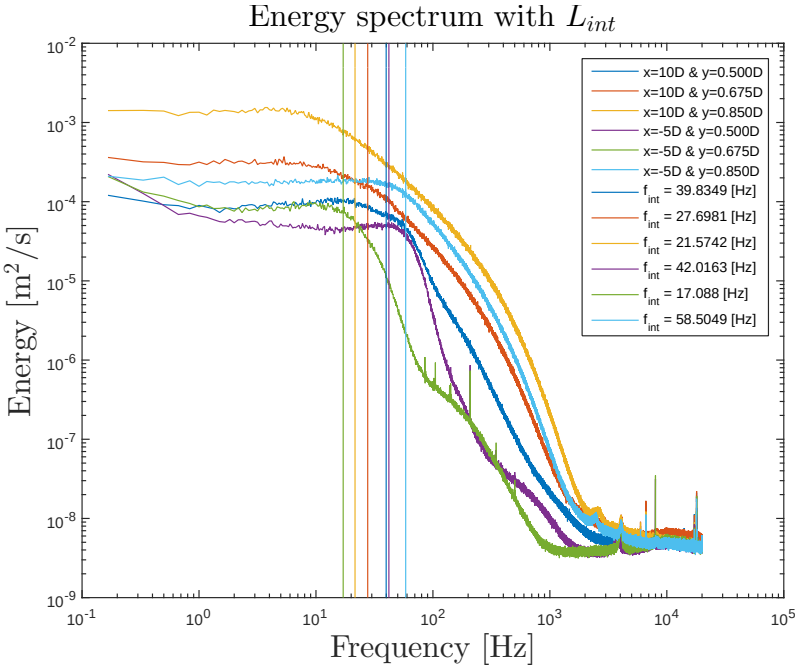


Figure 38: Energy spectrum with L_{int} expressed as f_{int}

Here it can be seen, that the integral length scales appear in the area around the first kink, as expected. After the lines of the integral length scales, the curve drops, and enters the inertial subrange. The curves in this range should follow the $\frac{-5}{3}$ rule of Kolmogorov. An energy spectrum with the $\frac{-5}{3}$ slope has been plotted together with the curves. This can be seen in Figure 39.

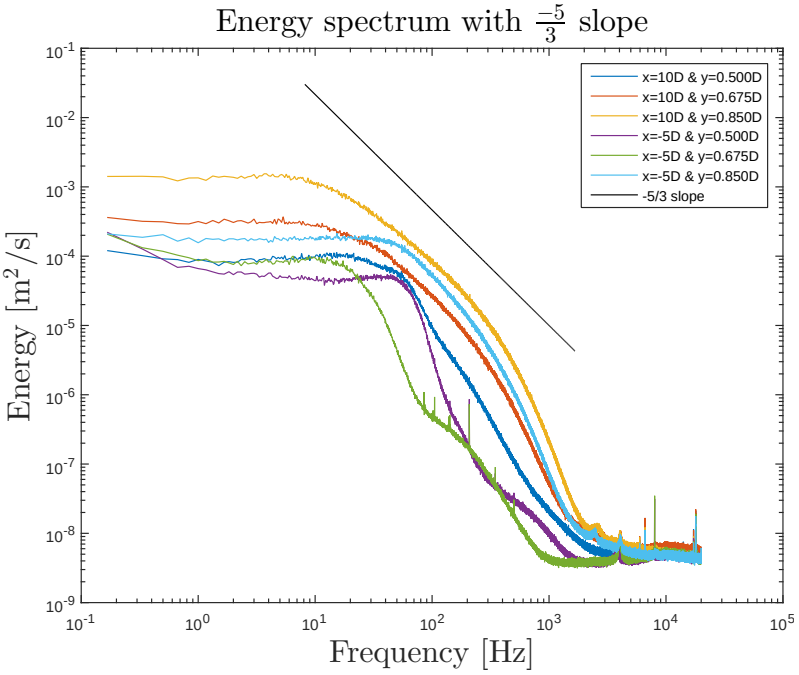


Figure 39: Energy spectrum with $\frac{-5}{3}$ slope

It can be seen, that some of the curves follow the rule. But the curves for the -5D position at 0.500D and 0.675D and for the 10D potion at 0.500D do not follow the $\frac{-5}{3}$ slope. The energy spectrum is only valid for measurements with sufficiently high RTI, which in these points are low. This indicates that the energy spectra in these points does not follow the assumption, and the curves will therefore not follow the $\frac{-5}{3}$ slope. In the remaining measuring points, the curves follow the slope and with a higher RTI, the assumption is fulfilled.
Also in the inertial subrange, the Taylor microscale is expected to be found. The Taylor microscale was converted into frequencies via Equation (54) and was plotted together witht the energy spectra in Figure 40.

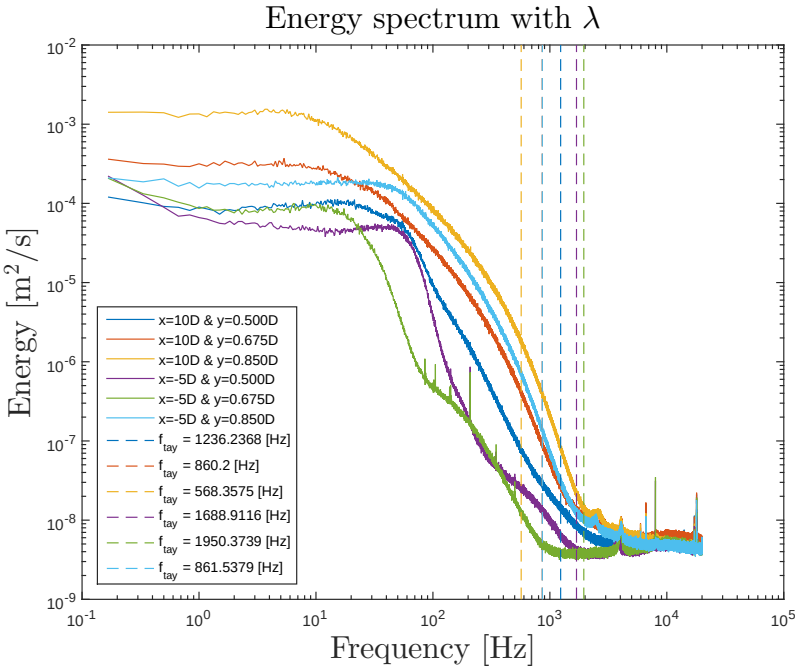


Figure 40: Energy spectrum with λ expressed as f_{tay}

The Taylor microscales for the 10D position at 0.675D and the -5D position at 0.850D are almost similar, and the lines are therefore overlapping. Further more, it can be seen that the Taylor microscales for the curves from before (-5D position at 0.500D and 0.675D and for the 10D potion at 0.500D) seem out of place, but that can be expected as the curves obtained do not fulfil the assumption of high RTI. But for the remaining curves it can be seen, that the lines intersect with the curves within the inertial subrange. For the curve at the 10D position at 0.850D, this is especially pronounced, since the line here represents the ending of the linear slop and thereby indicates the transition between the inertial subrange and the dissipation subrange.
The next step of the curve is representing the dissipation subrange. It can be seen, that the curve after the Taylor microscale is decreasing faster, indicating that the viscus forces are dominating and the turbulent kinetic energy is dissipating into heat. It can be seen, that the curves at high frequencies flatten out. This indicates that the natural limit of how high frequencies the hot-wire can measure has been reached. The flat part appears at very low energy levels and the peaks seen is most likely noise. This also means, that the last part of the dissipation subrange is shown in these energy spectra. The Kolmogorov length scale is representing the smallest eddies, so the scale can be expected to be within this flat part of the curve. The Kolmogorov time and length scales were calculated from the dissipation rate via Equation (49), and the correlations Equation (50) and Equation (51). The viscosity of the air was found for each point from the recorded ambient temperature and pressure[1]. The results can be seen in Table 5.

[1]The kinematic viscosty was found via the software EES: nu=KinematicViscosity(Air_ha;T=T_atm;P=p_atm)

x-position	y-position	$\varepsilon[\frac{m^2}{s^3}]$	$\nu \cdot 10^{-5}[\frac{m^2}{s}]$	$T_{kol} \cdot 10^{-3}$ [s]	$L_{kol} \cdot 10^{-4}$ [m]
10D	0.500D	6.457	1.544	1.546	1.545
10D	0.675D	6.919	1.544	1.494	1.519
10D	0.850D	10.565	1.544	1.209	1.366
-5D	0.500D	7.171	1.542	1.466	1.504
-5D	0.675D	6.629	1.550	1.529	1.540
-5D	0.850D	8.871	1.549	1.321	1.431

Table 5: Kolmogorov time and length scales

The Kolmogorov time and length scales have been plotted in Figure 41.

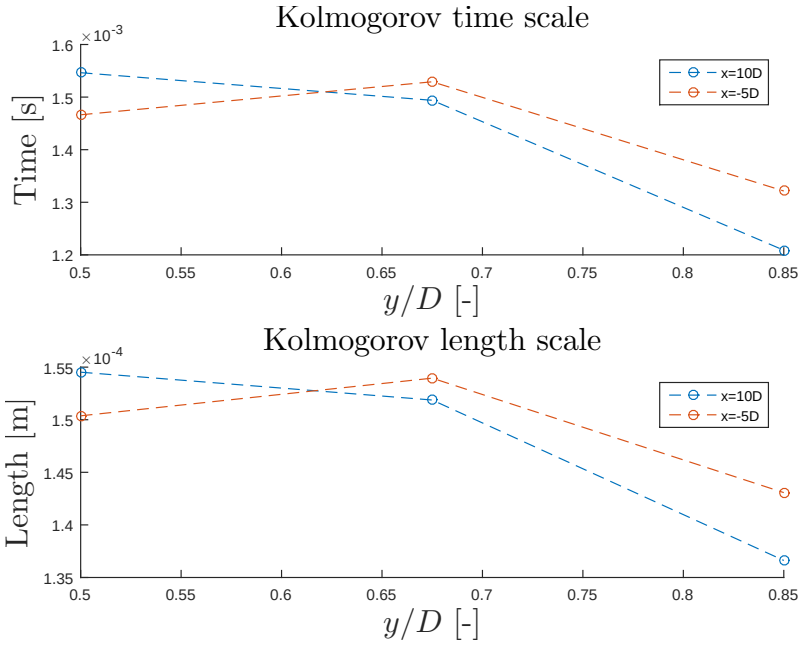


Figure 41: Kolmogorov time and length scales

The difference between the time and length scales are small. But a decreasing tendency for the 10D position can be seen. It can be expected, that the life time of the eddies are smaller near the wall, because of increased friction from the walls. The RTI is also higher at the walls, so the size of the eddies can be expected to be smaller here. At the -5D position, the tendency follows the one for the integral time and length scales, by having a peak at 0.675D. The RTI is low here, so the time and length scales can be expected to be bigger.
The Kolmogorov length scales have been transformed into frequencies via Equation (53) and plotted in the energy spectra, Figure 42.

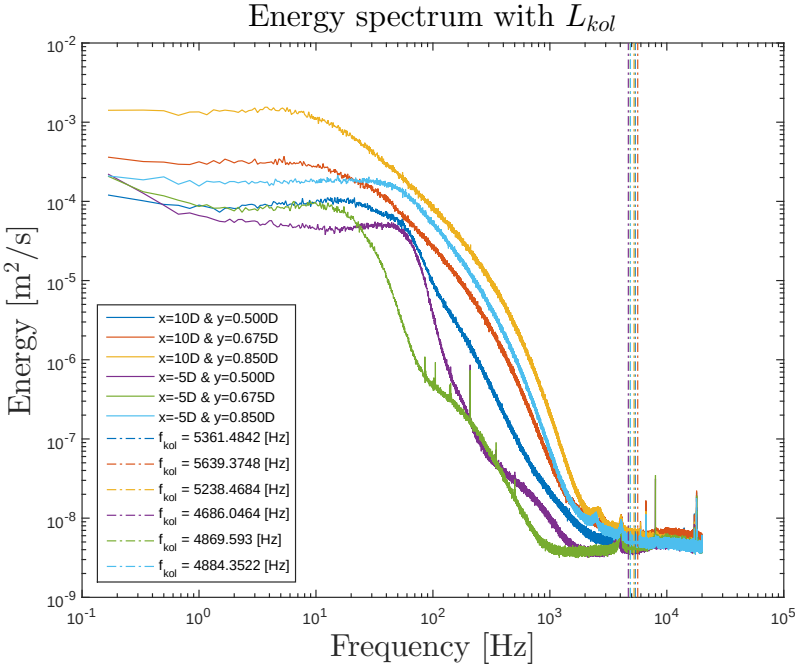


Figure 42: Energy spectrum with L_{kol} expressed as f_{kol}

Here it can be seen, that the Kolmogorov length scales are close to each other, and they are located in the flat part of the curve as expected. In order to obtain an energy spectrum which can detect the Kolmogorov length scales, a much smaller wire and an isolated system to prevent noise have to be used.

With this, the integral and Kolmogorov time and length scales, and the Taylor microscale have been obtained. A plot with all scales shown at once can be seen in Appendix E.

5 Engine test set-up

The equipment used to carry out the engine tests and the test procedure will here be presented. The purpose of the engine test is to make preliminary measurement of the time and length scales in an IC engine, in order to understand the difficulties regarding turbulence measurement. From the measurement, a basis for measuring turbulent structures in an IC engine will be established.

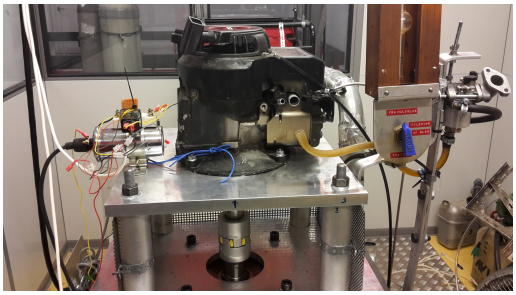
5.1 Engine

The engine used for the experiments was a Briggs and Stratton, 1 cylinder, 4-stroke SI IC engine. The data of the engine can be found in Table 6.

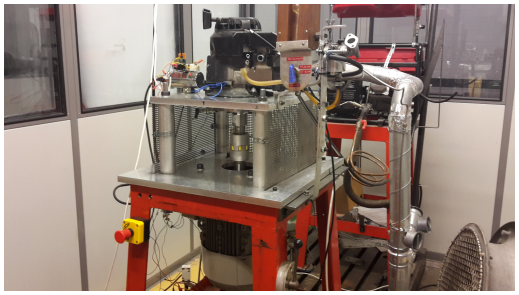
Parameter	Symbol	Value	unit
Stroke	S	0.06508	m
Bore	B	0.04765	m
Crank shaft arm length	R	0.02382	m
Connection rod	L	0.08890	m
Compression ratio	ε	6	-

Table 6: Engine data

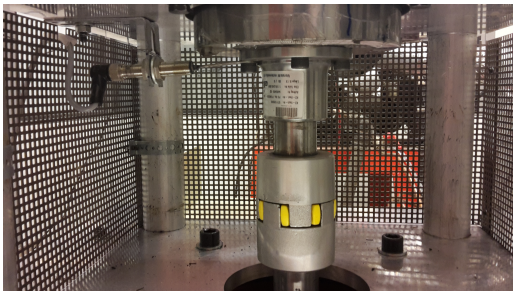
Because the experiments have to be conducted without combustion to protect the hot-wire, the engine was driven by an electric motor. The electric motor was controlled by a frequency converter through the software "Starter". The engine speed was set by "Starter", and could be varied between 200 and 3500rpm. In order to know the position of the crank an inductive sensor was fitted to the shaft of the electric motor. The sensor gives a pulse for each revolution. The sensor was not mounted such that the electric signal gave a pulse at TDC or BDC, but was placed 135° ATDC. This had to be accounted for in the data processing.
Pictures of the engine set-up can be seen in Figure 43. A picture of the mounting of the inductive sensor, and the geometry of the cylinder head can be seen.



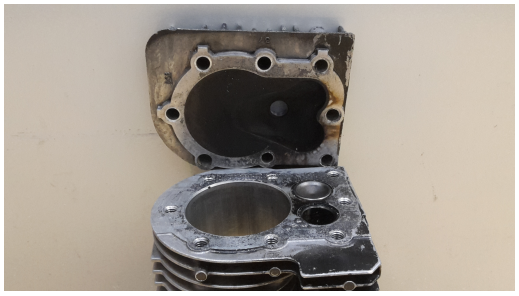
(a) Engine set-up



(b) Engine set-up



(c) Inductive sensor set-up



(d) Cylinder head

Figure 43: Engine set-up

5.2 Equipment

The elements used in this test set-up, are described via a measuring chain as before. The chain contains of the same elements as the pipe test, seen in Figure 23.

5.2.1 Hot-wire probe, mount and wiring

The hot-wire used in the engine experiment is a 55A75 high-temperature probe from Dan-Tec. The probe has a wire with a length and diameter of $2.2mm$ and $10\mu m$ respectively. The wire material is made of 90 % platinum and 10 % Rhodium. The probe can operate in temperatures and pressures up to $750°C$ and $100Bar$ respectively.
Because the hot-wire system has to cope with the high temperatures and pressures, the hot-wire is build into the wire support, and can therefore not be separated as in the pipe test. The support is a thin walled pipe, with an outside diameter[2] of $4mm$, filled with a ceramic material in order to isolate the electric wires inside. The probe is connected to the miniCTA with a 4m cable.

In order to mount the hot-wire in the engine, a fixture had to be made. It was decided to use a Swagelok tube fitting[3] which can be seen in Figure 44.

[2]Follows DIN2391 part 1

[3]http://swagelok.com/en/catalog/product/detail?part=SS-4M0-1-2RPBT. Swagelok tube fitting, Part No. SS-4M0-1-2RPBT.

Figure 44: Swagelok tube fitting

The thread of the fitting is a 1/8 in. male thread, which does not fit the M14x1.25 female thread in the spark plug hole of the engine. A thread adapter was designed and produced in the workshop. The adapter was designed to fit the Swagelok fitting in one end and to imitate a spark plug in the other. A picture can be seen in Figure 45 and the technical drawing can be seen in Appendix F.

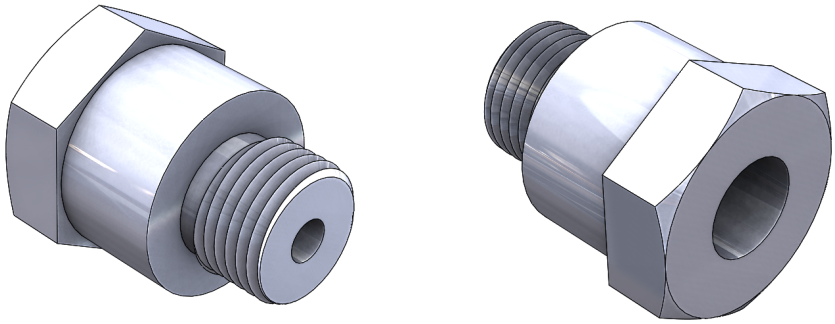


Figure 45: Thread adapter (front and back)

One washer was put between the fitting and the adaptor, and one was put between the adaptor and the engine to make a sealed connection.

The fixture was tested in order to assure a proper mounting of the hot-wire, meaning that the hot-wire is not blown out of the engine under operation and the wire support is not damaged. A dummy-support has been specially produced by DanTec, which is identical to the wire support used with the hot-wire. The dummy-support was used to test the fixture without risk of breaking the hot-wire. An estimation of the amount of force, that the fixture has to withstand was made. An engine pressure of $100bar$ was assumed,

which will lead to a force along the wire support. The force was calculated as:

$$F = p \cdot A = p_{engine} \cdot \frac{\pi}{4} D^2_{support} = 100 \cdot 10^5 Pa \cdot \frac{\pi}{4} \cdot \left(4 \cdot 10^{-3} m\right)^2 = 125.7N$$

leading to

$$m = \frac{F}{g} = \frac{125.7N}{9.82 \frac{m}{s^2}} = 12.8kg$$

Only weights of $12kg$ could be found, corresponding to an engine pressure of $93.8bar$. The resistance of the dummy-support was measured and the Swagelok fitting was mounted. The Swagelok fitting was tighten after the recommendations from Swagelok. With a $4mm$ fitting the nut should be finger tight and then turned additional 270°. For smaller tube fittings 450° is recommended, and because the $4mm$ fitting is on the borderline of being a small fitting, it was decided to tighten the nut an additional 360° from finger tight. In order to insure an easy dismounting of the fitting, teflon ferrules where used with the fitting. The resistance through the dummy-support was measured, and no change was observed. Meaning that the tightening has not damaged the support. The dummy-support with the mounted fitting was tested with the load of $12kg$ and no movement between the two parts was observed. This means that the fixture will be able to sufficiently hold on the the support. This load is static, so the fixture was tested in the engine in order to assure that the fixture was sufficient with a pulsating load. The engine was run at engine speeds from 600 to $3500rpm$ and no movement of the support was observed. With this, it was concluded that the fixture was able to withstand the engine pressure without damaging the support. The dummy-support and the dummy-support with an exploded view of the fixture can be seen in Figure 46.

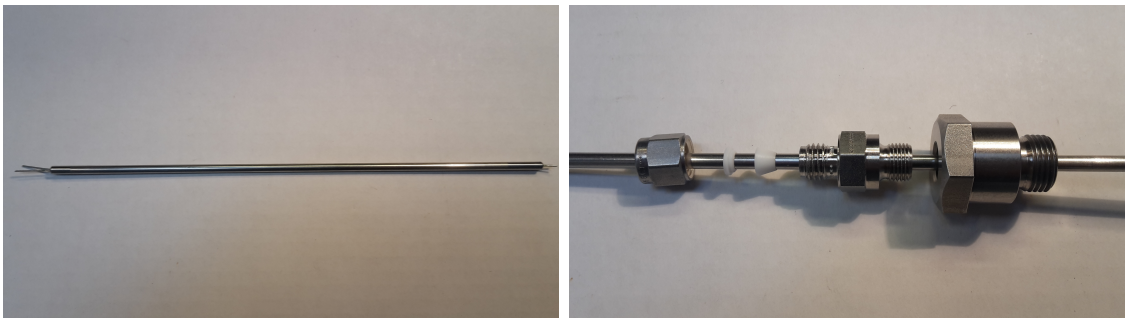


Figure 46: Dummy-support with fixture

5.2.2 MiniCTA system

The MiniCTA system used in this experiment was a MiniCTA 54T42 from DanTec Dynamics. This is a newer version of the 54T30 miniCTA used in the pipe test. DanTec was concerned about using the old version together with the high-temperature probe because the wire is thicker than the 55P11 probe, and due to the start up of the 54T30 miniCTA, the wire had a risk of breaking. The 54T42 miniCTA was made to cope with this start up problem, and therefore used in this experiment. The functions and layout of the 54T42 miniCTA is exactly the same as for the 54T30 miniCTA, as seen in Figure 47.

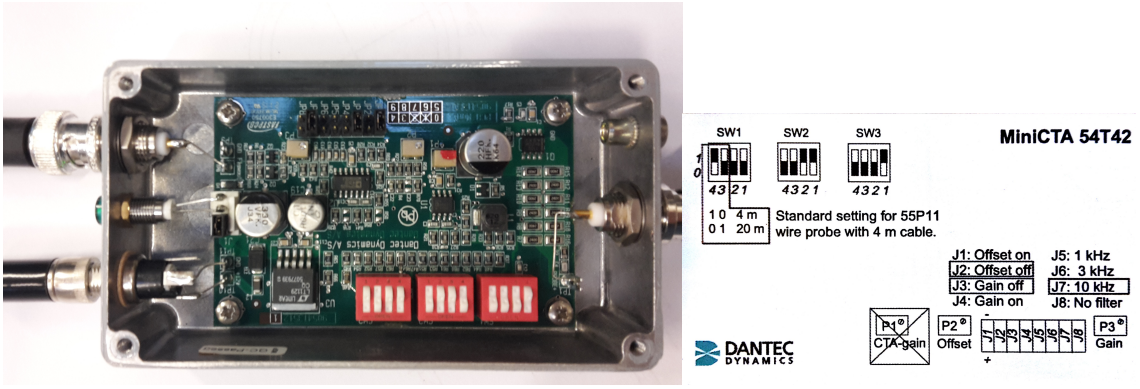


Figure 47: 54T42 miniCTA during measurements and its options

For the test, the gain was switch off, the offset was switched on and a low-pass filter of $10kHz$ was chosen.

5.2.3 A/D module

For the engine test, a NI 9215 with BNC, A/D module was used. It was connected via a Hi-speed USB carrier to a laptop. The module is a 16-bit module with four channels and a range of $\pm 10V$ and a maximum sample frequency of $100kHz$. The hot-wire was connected to channel 0 and the inductive sensor connected to channel 1. The module has a build in bias resistor of $200k\Omega$ between each channel. It was found that the bias resistor was too big, making the signals drift too much apart and thereby disturbing the signal. The build in bias resistor was bypassed by connecting the two channels with a smaller resistor of $10k\Omega$. In this way, the signals were brought closer together and noise transferred between the channels has been minimised.

5.2.4 Software and data acquisition

The "StreamWare Basic" software was again used to set-up the miniCTA and make the calibration. Because temperature correction has to be performed, only the calibration points were saved from the calibration, and were used later in the data processing.
The data acquisition was in these experiments performed via MatLab. When performing the data acquisition, only the raw voltage signal from the hot-wire and the induction sensor with the matching time stamp were saved in a .mat file. The MatLab script used to make the data acquisition can be seen in Appendix G.

5.3 Test procedure

The test procedure is here presented, and describes the order and method of how the set-up, calibration and measurement was carried out. Before the engine tests were performed, pre-tests was made in order to design the real tests and determine parameters of the set-up.
First the hardware set-up was made and the support fixture was mounted. The adaptor and Swagelok fitting was connected with a washer in between. The adaptor and fitting was mounted on the wire support as described. The hot-wire with the fixture was mounted in

a spare cylinder head in order to adjust the depth and angle of hot-wire. The parts in the measuring chain was connected, with the power switched off. The measuring chain was entered in the "StreamWare Basic" software and the overheat ratio was set to 0.8. The miniCTA was set up by turning the gain off and the off-set on. During pre-test it was seen that the volt signal was idling at around $3V$, and during operation exciting $5V$. With the build in off-set switched on, the volt signal was idling around $2V$ and was kept under $5V$ during calibration. The low-pass filter was set to its highest setting of $10kHz$. The overheat ratio was adjusted in the miniCTA from the instructions in the software. The probe was placed in the stream of the jet and the power for the miniCTA was switched on.
The hot-wire was calibrated via the software, in the range which were possible to do with the jet. A test showed that the jet could produce wind speeds from $1.3\frac{m}{s}$ to $28\frac{m}{s}$, so this range was used as the calibration range. The 10 calibrations points was saved in an .txt file and was used later in the temperature correction in the data processing. The wind speed of the jet was again found from the micro manometer and Equation (63). The calibration procedure followed the procedure used to calibrate the hot-wire in the pipe test.
The engine was started and the inductive sensor was switched on. In order to start up the engine, the software "Starter" was started and set up according to the original set-up, [3]. First the parameters were set to match the values in Table 7.

Parameter	Description	Set value	Unit
	Motor Start		
p10	Drive commissioning parameter filter	[0] Ready	-
p1900	Motor data identification and rotating measurement	[0] Inhibited	-
p1910	Motor data identification selection	[0] Inhibited	-
	Motor Speed		
p2000	Reference speed reference frequency	[3500]	rpm
p1082	Maximum speed	[3500]	rpm
	Brake usage		
p1240	Vdc controller configuration	[0] Inhib Vdc ctrl	-
p219	Breaking resistor braking power	0.93	kW
	Data Tracing		
p45	Display values smoothing time constant	1000	ms

Table 7: "Starter" parameters for frequency converter

Once the parameters had been set, the relay outputs were set such that relay 0 should be open through COM and NC 0 in ports 20 and 18. Relay 1 should be open through COM and NC 1 in ports 22 and 21. With the relays set, the engine speed can be typed in and the engine was ready to start. Before the engine was started, the hot-wire was moved, still without disconnecting it, and mounted in the engine. The engine was started and the signal of the hot-wire was controlled via the software "NI MAX". The data acquisition MatLab script was prepared by entering the engine speed, and the name of the measurement performed. The data acquisition was set to measure with a sample rate of $100kHz$ in order to assure to capture the pulses from the inductive sensor. In order to again obtain a signal with a variability of 5%, 400 engine cycles were recorded. But because the signal need to be shifted with $135CAD$ and processed extra cycles were

recorded. From the sample rate and the engine speed, the sample duration, SD, could be calculated. With this, a continuous voltage signal from the inductive sensor and the hot-wire was obtained. Because the signal had to be cut up into individual cycles and only the closed engine rotations can be used, there was no need for making 400 individual measurement as in the pipe test.
Data about the experiment and the datafile name were written in an Excel document, which can be seen in Appendix H.
The test was performed for 4 positions and 3 engine speeds. All measurement have been conducted with the throttle fully open. The variation of test parameters can be seen in Table 8.

Parameter	**Settings**	**Unit**
Depth	[4 ; 8]	mm
Angle	[0 ; 90]	deg
Engine speed	[500 ; 1000 ; 1500]	rpm

Table 8: Variation of test parameters

The depth is measured from the cylinder head and towards the piston. The angle of the hot-wire is described in Figure 48.

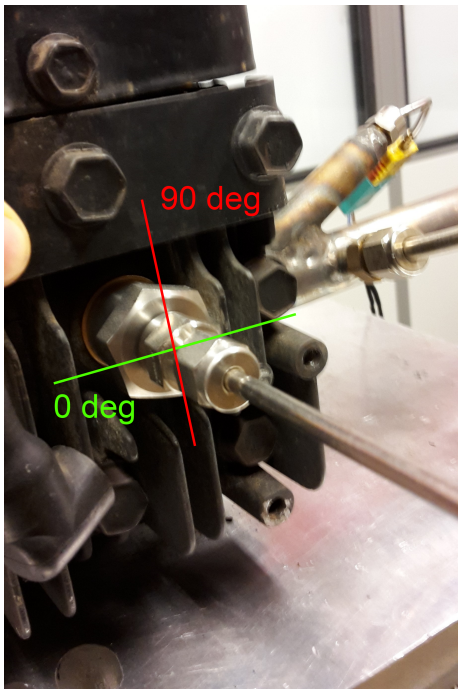


Figure 48: Angle of hot-wire in engine

When the position of the hot-wire was fixed via the fixture, the engine speed was varied. Then the engine was stopped and the hot-wire removed. A new position was set and the wire was again installed in the engine. It was discovered that when the hot-wire was installed, the cable from the probe support could get twisted, leading to a raised voltage signal. By making sure that the cable was not twisted, the voltage level dropped to a

normal level.
The atmospheric temperature and pressure were varying within $0.4°C$ and $118Pa$, during the test and were therefore assumed constant.

When all measurement were performed, the data processing could begin. The same method of dividing the data, in order to not avoid running out of RAM, was used as for the pipe test. The data were divided into three categories, one for each engine speed. The data processing MatLab script made for the engine test is based on the one from the pipe test. But because temperature correction and cutting of the velocity signal had to be performed, the script was extended in order to obtain the velocity signal which could then be fed into the old part. The new part of the script cut the continuous voltage signal into engine rotations via the pulses from the inductive sensor. The pulses were corrected with the $135CAD$ which the inductive sensor was shifted from TDC. The rotations which contained the closed rotation of the engine cycle were stored. Next the simulation from Section 2.6 for obtaining the pressure and temperature in the engine as function of CAD was performed. With that, the temperature correction from Section 2.1.3 was performed and thereby also the velocity signal. Before the processing of the signal was performed, the signal was plotted to check if the right engine rotation was used. If the wrong rotation was used, the cutting of the signal was changed such that the right rotation was used. From the simulation, an average viscosity over the cycle was found and was used when calculating the Kolmogorov scales. With these two corrections, the script was run fully and a data base was created. The MatLab script with associated functions can be found in Appendix I. A script collected and plotted the results. The script can be seen in Appendix J.

6 Engine test

The result of the engine test is here presented. The flow is described in the same manner as in the pipe test, with average velocities and relative turbulent intensity to get a feeling of the flow. Followed by calculation of the integral scales from the autocorrelation function together with the Taylor microscale. And last the energy spectrum and the Kolmogorov scales. These steps are made in order to see if the behaviour of the engine can be reflected in the obtained results.
To make use of hot-wire anemometry in an engine is more complicated than making measurement in a simple pipe flow, so some simplifications are made.

1. The measurement are conducted using a single wire hot-wire. In a highly 3 dimensional flow, this will make the hot-wire more influenced by flows with other direction than intended. When using a 1D probe to make 2D measurement, the two velocities will be higher because the compensation normally used with a 2D wire is not made.

2. The temperature used to make the temperature correction was conducted from a simple simulation, with the assumption of isentropic process.

3. The equations used for make the temperature correction are only taking convection into account. Conduction to the prongs and radiation are neglected.

4. An average temperature of the viscosity was used when calculating the Kolmogorov scales.

During the test, some problems with the inductive sensor occurred. This was discovered when analysing the data. For the measurements at $1000rpm$ and $1500rpm$, the sensor only recorded every second rotation, equivalent to a full engine cycle. In order to solve the problem, a averaging between two pulses was made in order to be able to cut the voltage signal into single rotations. The amount of points between two pulses was found, and within the individual 400 engine cycles, the point was varying with less than half a percentage. So it was assumed that manual introducing a new point in the voltage signal of the inductive sensor would not change the outcome of the results. For the measurement performed at a depth of $4mm$, $90deg$ and $500rpm$ the signal from the inductive sensor was too disordered to use, so this measurement has not been taking into account in the following results.

6.1 Sensitivity analysis

A sensitivity analysis of three areas of the calculations have been made in order to see their effects on results. The results presented in this section is not final and are only used in order to describe the changes of the results of this analysis. An explanation of the derivation of the engine results will be explained in the following sections. An example of a velocity signal which will be used in this section can be seen in Figure 49.

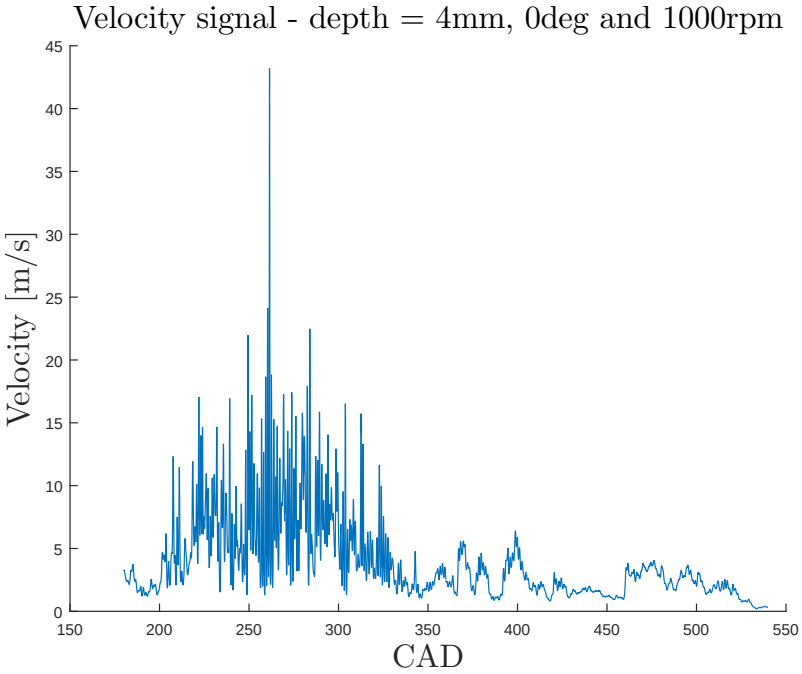


Figure 49: Velocity signal obtained at $depth = 4mm$, $0deg$ and $1000rpm$

6.1.1 Polynomial fit VS. high-pass filter for obtaining velocity fluctuations

From the pre-test it was discovered that the fluctuations of the velocity signal could not be obtained by subtracting the average velocity from the velocity signal due to the cyclic behaviour of the engine cycle. For the pre-test a polynomial fit was subtracted from the velocity signal in order to obtain the fluctuations. But a proper fit were difficult to obtain, so a high-pass filtering were carried out instead. Result for both method were calculated in order to see the difference. The polynomial fit was a third order polynomial and the high-pass filters cut-off frequency was put to $200Hz$ on basis of the pre-test. From Figure 50, fluctuations from a velocity signal obtained from a measurement at $depth = 4mm$, $0deg$ and $1000rpm$, for interval 1, can be seen for both methods.
Here it can be seen that the curve for the polynomial fit start out being lower than the cure for the high-pass filter. After some time the two curves crosses and the curve for the polynomial fit is en general higher in the last half of the figure. It seems like the polynomial fit does not lower the last part of the curve sufficiently, and that the high-pass filter separate the fluctuation from the velocity signal in the best way.

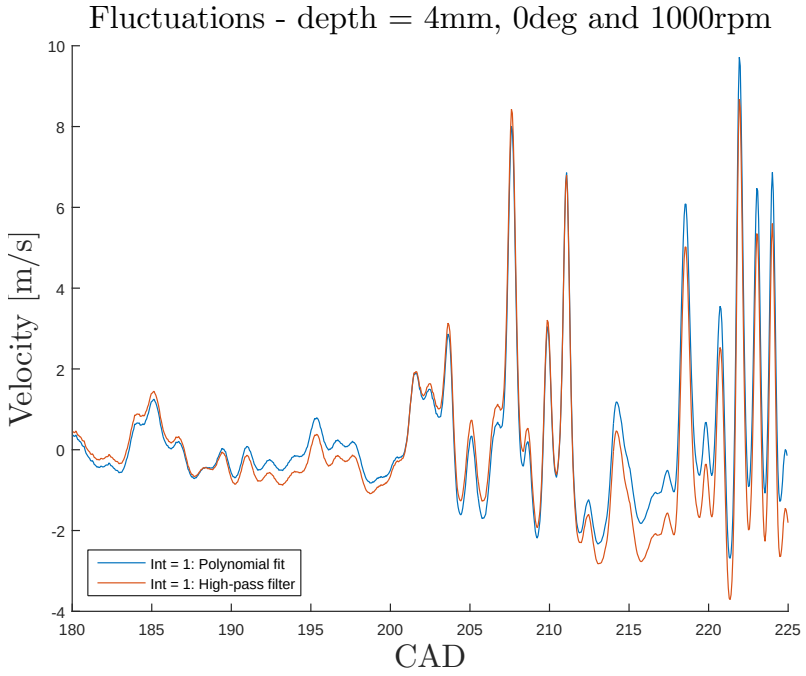
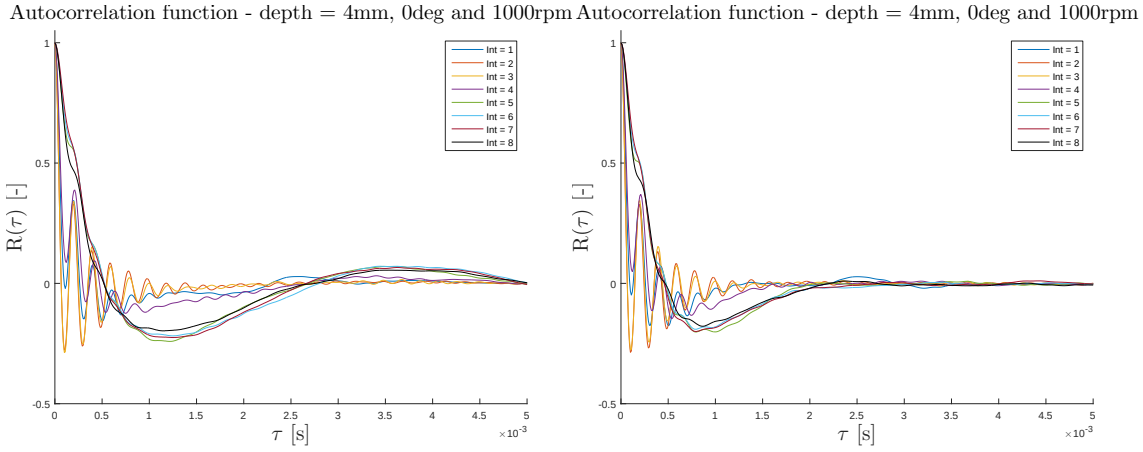


Figure 50: Variation of velocity fluctuations with use of polynomial fit and high-pass filter. $depth = 4mm$, $0deg$ and $1000rpm$

Effects of using a high-pass filter can also be seen in the autocorrelation functions. At the same position, but including all the 8 intervals, the autocorrelation functions can be seen in Figure 51.



(a) With polynomial fit

(b) With high-pass filter

Figure 51: Variation of the autocorrelation function with use of polynomial fit and high-pass filter. $depth = 4mm$, $0deg$ and $1000rpm$

Then main effect of using a high-pass filter is that the biggest eddies are filtered out, which can be seen on Figure 51[b], since the autocorrelation function stabilizes around 0 faster. Using a high-pass filter with a too high cut-off frequency can lower the integral

time and length scale because the biggest eddies are removed.
An energy spectrum has been conducted and can be seen in Figure 52.

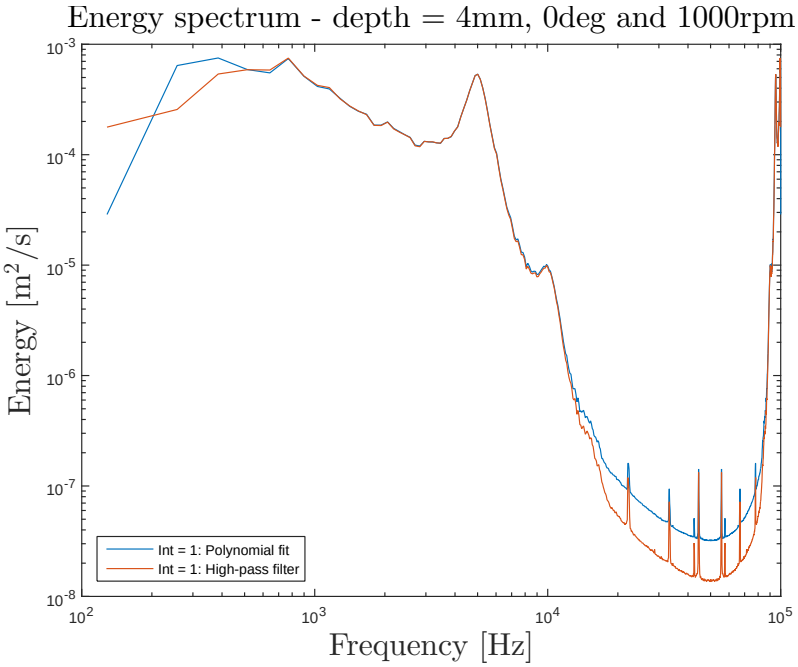


Figure 52: Variation of spectrum with use of polynomial fit and high-pass filter. $depth = 4mm$, $0deg$ and $1000rpm$

From the first part of the spectrum it can be seen, that with the high-pass filter, the energy at low frequencies are more constant than with the polynomial fit. With high-pass filter the curve look more as result obtained from a one dimensional measurement, which could indicate the the contamination from flows with other directions have been excluded. In the middle part of the spectrum are the curves identical, but in the last part they differ again with the curve with the high-pass filter having a lower level of energy. The results after $10kHz$ cannot be analysed properly because of the low-pass filter.
The time and length scales have been calculated and can be seen in Table 9

Scale	**Polynomial fit**	**High-pass filter**
$\boldsymbol{T_{int} \cdot 10^{-5}}$ $[s]$	5.043	4.926
$\boldsymbol{L_{int} \cdot 10^{-4}}$ $[m]$	1.957	1.911
$\boldsymbol{\lambda \cdot 10^{-5}}$ $[m]$	8.144	8.090
$\boldsymbol{T_{kol} \cdot 10^{-5}}$ $[s]$	1.093	1.095
$\boldsymbol{L_{kol} \cdot 10^{-5}}$ $[m]$	1.036	1.037

Table 9: Scales with with use of polynomial fit and high-pass filter. $Int = 1$, $depth = 4mm$, $0deg$ and $1000rpm$

From the table it can be seen that there is almost no change in the Kolmogorov scales, when applying the high-pass filter, which was expected because only the biggest eddies have been excluded. The biggest change is at the integral scale, where the integral time scale for the one with the high-pass filter is lower than with the polynomial fit. This

can be expected because the biggest eddies might have excluded. This indicated that the cut-off filter maybe is set too high.
From this analysis it is concluded to use the high-pass filter instead of the polynomial fit, due to a more accurate separation of the fluctuations, and the change in the integral scales are very small.

6.1.2 Estimation of cylinder gas temperature

Another area which have been analysed is the effect of the temperature of the cylinder gas. In this study a very simple estimation have been carried out, and in order to see how sensitive the results are to temperature changes, the estimated gas temperature is here compared to a temperature found by artificial lowering the temperature pressure curve with 5%. Meaning that the pressure is kept unchanged. It can be expected that the temperature will fall if for example heat loss to the cylinder wall are taken into account. The velocity signal have been carried out for the two temperatures at the position at $depth = 4mm$, $0deg$ and $1000rpm$, and can be seen in Figure 53.

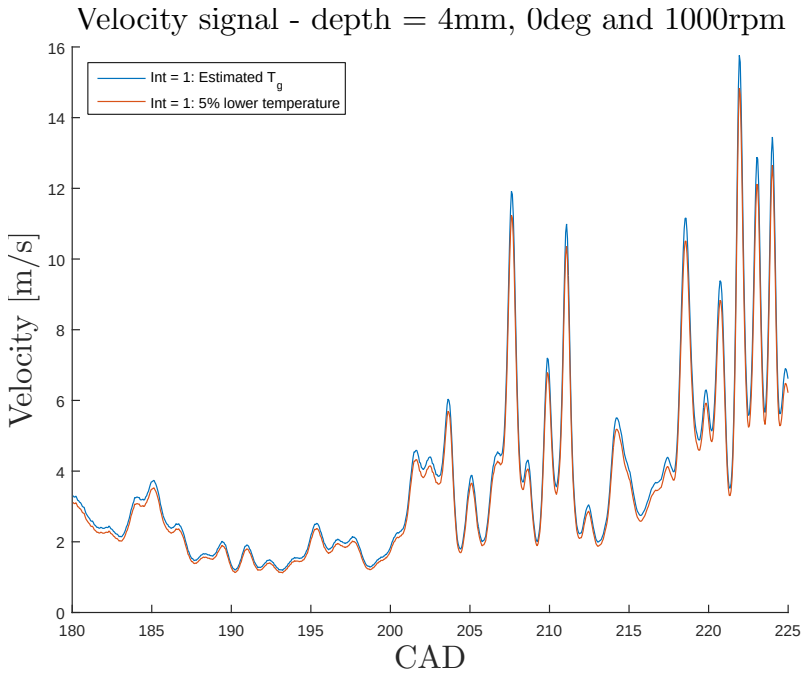


Figure 53: Variation of velocity with different gas temperature. $Int = 1$, $depth = 4mm$, $0deg$ and $1000rpm$

It can be seen that the lower temperature leads to a slight fall of the velocity. Because the curve have just been lowered, only very small changes can be seen in the autocorrelation functions, because the shape of the velocity signal would remain the same. So the autocorrelation function will remain almost exactly the same.
The energy spectrum have been conducted at the two temperatures and can be seen in Figure 54.

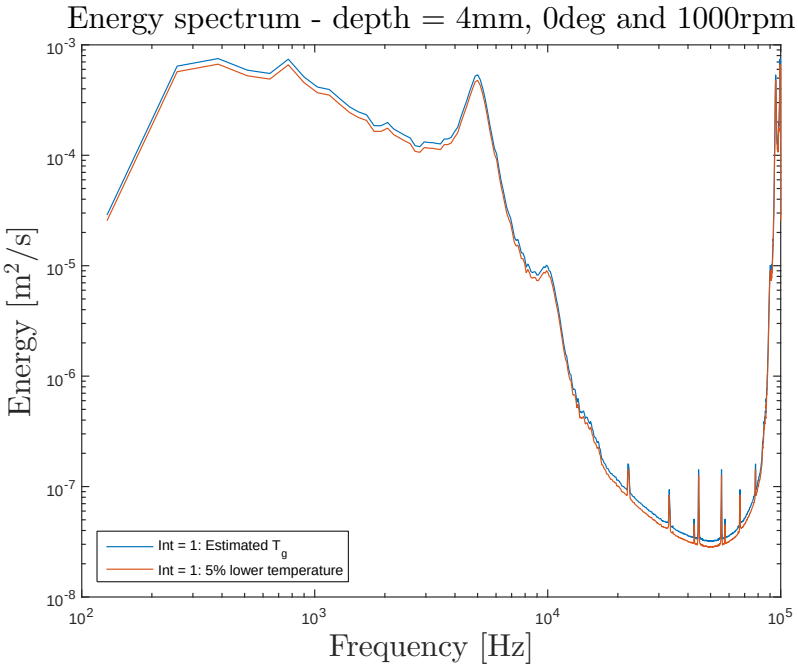


Figure 54: Variation of spectrum with different gas temperature. $depth = 4mm$, $0deg$ and $1000rpm$

It can be seen that the lower temperature leads to a slight fall of the energy. The small fall in the the energy can be due to slightly lower fluctuations, but is almost the same because the fluctuation have been separated.
The time and length scales a the two temperatures have been calculated and can be seen in Table 10.

Scale	**Estimated temperature**	**Lowered temperature**
$\boldsymbol{T_{int} \cdot 10^{-5}}$ $[s]$	5.043	5.044
$\boldsymbol{L_{int} \cdot 10^{-4}}$ $[m]$	1.956	1.843
$\boldsymbol{\lambda \cdot 10^{-5}}$ $[m]$	8.144	7.829
$\boldsymbol{T_{kol} \cdot 10^{-5}}$ $[s]$	1.093	1.114
$\boldsymbol{L_{kol} \cdot 10^{-5}}$ $[m]$	1.036	1.046

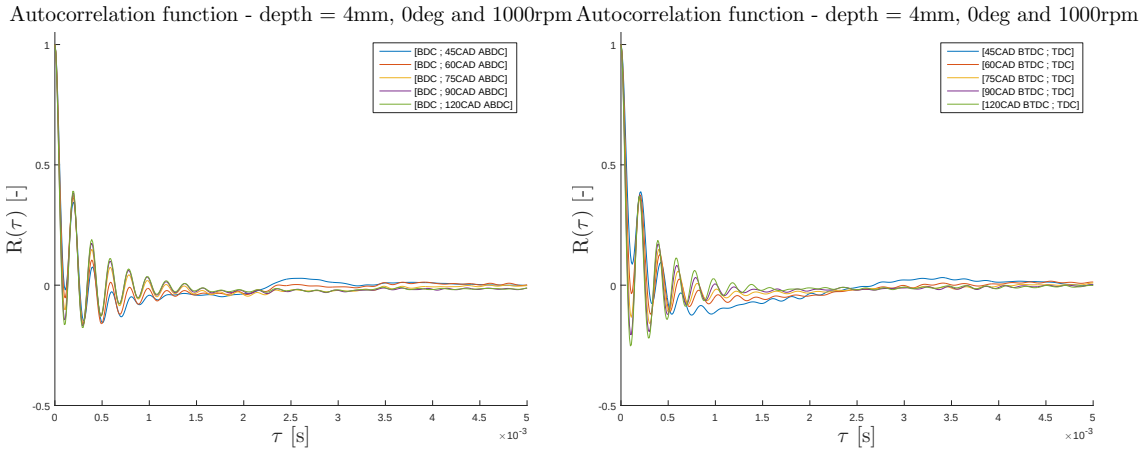
Table 10: Scales with different gas temperature . $Int = 1$, $depth = 4mm$, $0deg$ and $1000rpm$

From the results it can be seen that the integral time scale only differs slightly, which was expected due to the almost identically autocorrelation functions. The integral length scales differ due to the lower average velocity of the low temperature velocity signal. The same happens for the Taylor microscale, since the average velocity also is present in the calculation. The Kolmogorov scales are slightly raised, which can be from Equation (50) due to the higher viscosity and lower dissipation rate.

It can be seen that the temperature has an effect on the time and length scales, but the scales are not very sensitive. So it was concluded that the simple estimation of temperature was sufficient for this project, but can be improved for further work.

6.1.3 Length of intervals

The interval length was during pre-test set to $45CAD$. From Table 10 it can be seen that the integral time scale is around $5 \cdot 10^{-5}s$, and the duration of the $45CAD$ interval at $1000rpm$ is $750 \cdot 10^{-5}s$. This is equal to 150 times the integral time scale, and thereby less than the 400 necessary in order to avoid windowing. To see the effects of the short interval, the length of the interval have been varied. Two intervals have been made for the compression stroke; one from BDC and forward, and one from TDC and backwards. Four intervals length have been chosen, 45, 60, 75, $90CAD$. An interval length of $90CAD$ is equivalent to 300 integral time scales. An extra interval length equal to $120CAD$ have also been created because the interval length is equivalent to 400 integral time scales. The length of the interval will not have an effect of on the velocity, so only the autocorrelation function and the energy spectrum will be presented. The autocorrelation function for the 2 intervals as function of interval length can be seen in Figure 55.



(a) (b)

Figure 55: Autocorrelation function with different gas temperature. $Int = 1$, $depth = 4mm$, $0deg$ and $1000rpm$

From Figure 55(a), it can be seen that with longer intervals, the curves are oscillating stronger. This can be because the interval starts to stretch into a more turbulent area, which can cause the oscillations. The same trend can be seen from Figure 55(b)
The energy spectrum have been conducted in order to investigate the effects, and can be seen in Figure 56.

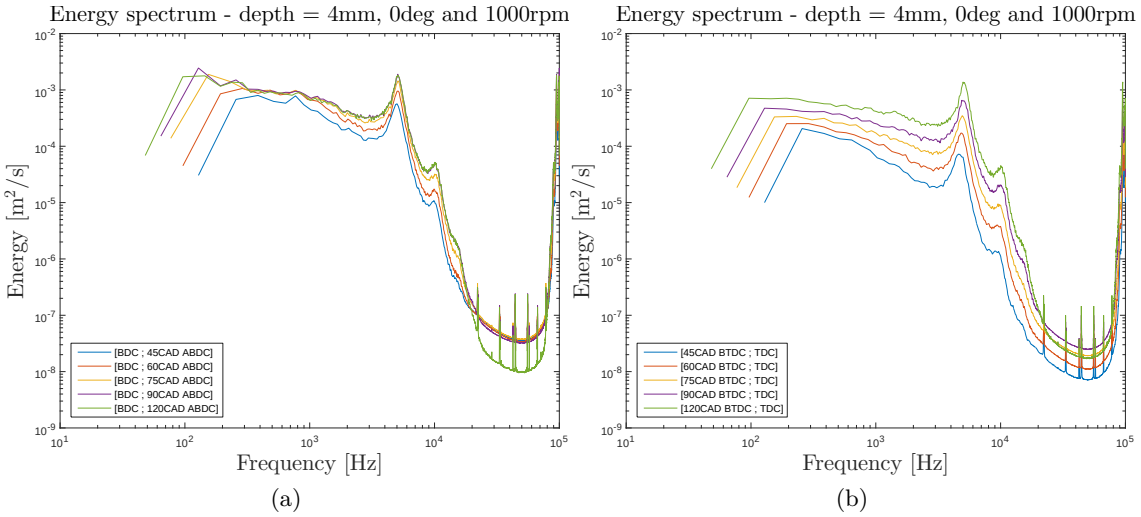


Figure 56: Energy spectra with varying interval length. $depth = 4mm$, $0deg$ and $1000rpm$

From Figure 56(a) it can be seen that the curves starts earlier and earlier as the interval gets longer. This shows that there are bigger eddies in the flow, which was not captured by an interval of $45CAD$. The curves are raised slightly as the interval gets longer, which can be due to the higher turbulent energy that can be found in the more turbulent are. The same trends can be seen in Figure 56(b). Here the curves also starts earlier and the energy level is raised. Again, it can be expected to find larger eddies when the interval is longer, and with a longer interval, it reaches into a more turbulent area.
The time and length scales have been found and can be seen in Table 11.

From BDC and forward					
Scale	**45 CAD**	**60 CAD**	**75 CAD**	**90 CAD**	**120 CAD**
$T_{int} \cdot 10^{-5}$ $[s]$	5.046	4.783	4.453	4.263	4.176
$L_{int} \cdot 10^{-4}$ $[m]$	1.955	2.190	2.201	2.187	2.095
$\lambda \cdot 10^{-4}$ $[m]$	1.121	1.258	1.253	1.182	1.015
$T_{kol} \cdot 10^{-5}$ $[s]$	1.506	1.372	1.176	1.025	0.886
$L_{kol} \cdot 10^{-5}$ $[m]$	1.216	1.160	1.074	1.031	0.933
From TDC and backwards					
Scale	**45 CAD**	**60 CAD**	**75 CAD**	**90 CAD**	**120 CAD**
$T_{int} \cdot 10^{-5}$ $[s]$	10.156	5.043	4.435	4.061	3.915
$L_{int} \cdot 10^{-4}$ $[m]$	1.920	1.091	1.129	1.128	1.491
$\lambda \cdot 10^{-5}$ $[m]$	2.134	2.584	3.259	4.243	6.464
$T_{kol} \cdot 10^{-6}$ $[s]$	7.351	6.483	6.056	5.990	6.729
$L_{kol} \cdot 10^{-6}$ $[m]$	8.495	7.978	7.710	7.669	8.128

Table 11: Scales with with varying interval length. $depth = 4mm$, $0deg$ and $1000rpm$

Looking at the integral time scale, it could be expected to see that the scale would grow when the interval got longer, but it can be seen that it actually decreases. Bigger eddies might be found by increasing the interval, but the integral time scale is a measure of the average live time for the biggest eddies. This means, that when the interval is

getting longer, bigger eddies are found, but the interval also stretches into a more turbulent are, where the integral time scale can be expected to decrease, and thereby bring down the average life time of the eddies. The integral length scale on the other hand are growing, but this is due to the higher average velocity. Looking at the autocorrelation function, from where the integral time scale is found, the oscillations disturbs the picture, and makes it difficult to obtain a proper result.
The Taylor microscale seems to fall when looking at the interval starting from BDC and growing when looking at the interval starting from TDC.
The Kolmogrov time scale seems to decrease for the interval starting at BDC, together with the Kolmogorov length scale. For the interval starting at TDC, the Kolmogorov time scale is more constant and so are the Kolmogorov length scale.

From this analysis, it can be seen that the length of the intervals has an effect on the time and length scales. However, it was decided to keep an interval length of $45CAD$ in the following results, because an interval length of $120CAD$ might include the largest eddies, but the interval is to long to see the changes which is coursed by the engine movement. So in order to resolve the movement of the engine, and the changes of the flow this brings, it was decided to keep the interval length to $45CAD$, but bearing in mind that the short interval disturbs the results of the time and length scales.

6.2 Average velocities and relative turbulent intensity

The average velocities of the flow have been found in 4 positions and at 3 engine speeds. Each measurement have been divided into 8 intervals of $45CAD$, representing the period from $180CAD$ to $540CAD$. The average velocities at the different positions as function of engine speed can be seen in Figure 57.

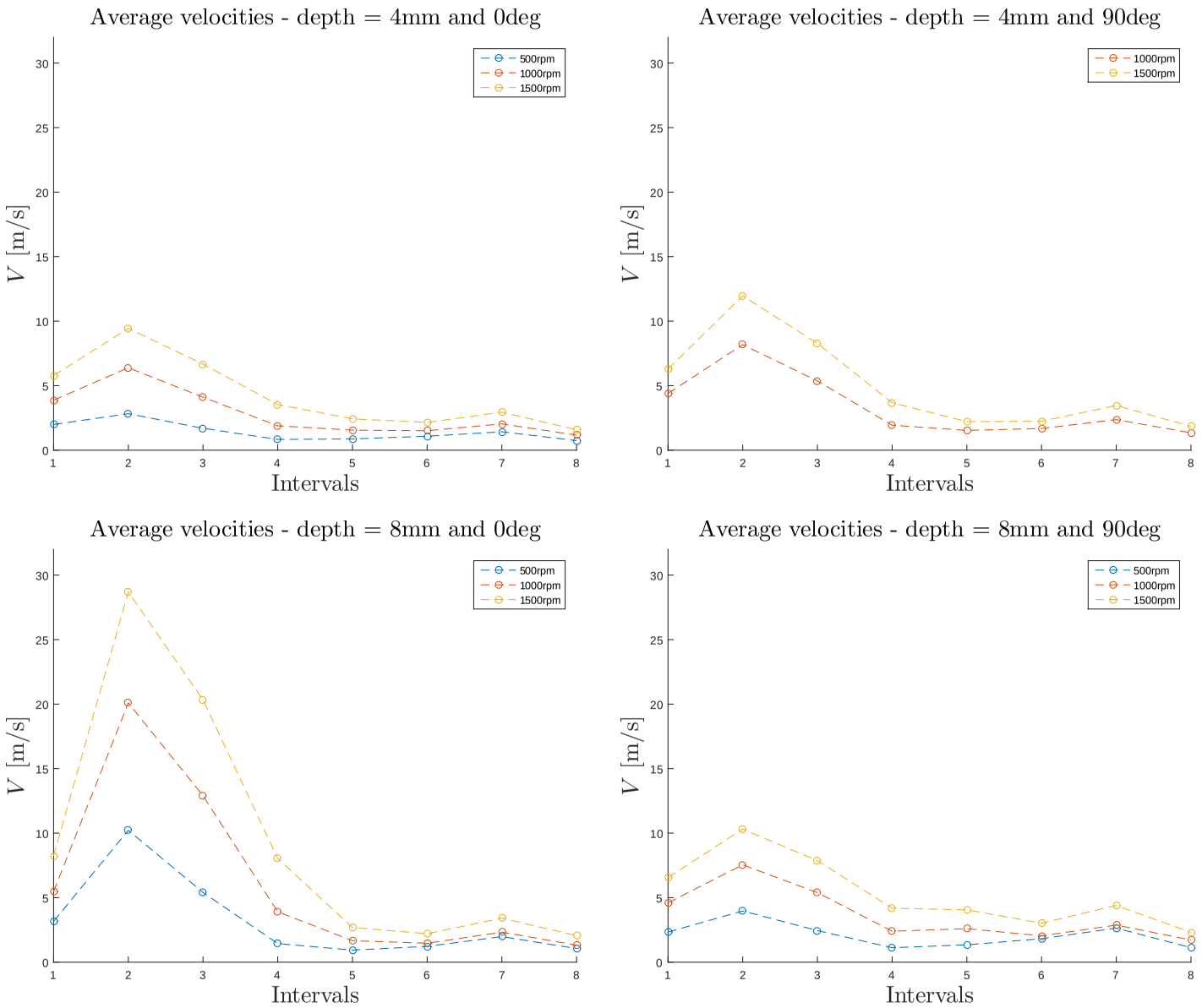


Figure 57: Average velocity with varying engine speed

The same tendency can be seen for all the positions. The highest velocities are found during the compression stroke, with maximum occurring during the second interval ($225CAD - 270CAD$). After TDC, during the expansion stroke, the velocity seems to remain constant at around $2 - 5\frac{m}{s}$. It can be seen, that higher engine speed leads to higher velocities, which also can be expected because the piston travels faster at higher rpm. Looking at the change in depth, it does not seems like the it has a big influence in the average velocity, but a small raise can be observed. The change of angle leads in general to a small raise in velocity. This is the same tendencies observed in [14][fig. 3]. For the $8mm$, $0deg$ position, it can be seen that the velocity is much higher than in the other positions. It looks like something is happening in this direction, but it seems peculiar as the velocity is much lower in the $4mm$, $0deg$ position. Looking the the Excel data sheet for the engine test, a comment about the twisting of the wire of the support can be seen. This have been mended before the test was carried out, but might still have an effect.

The relative turbulent intensity has been calculated can be seen in Figure 58.

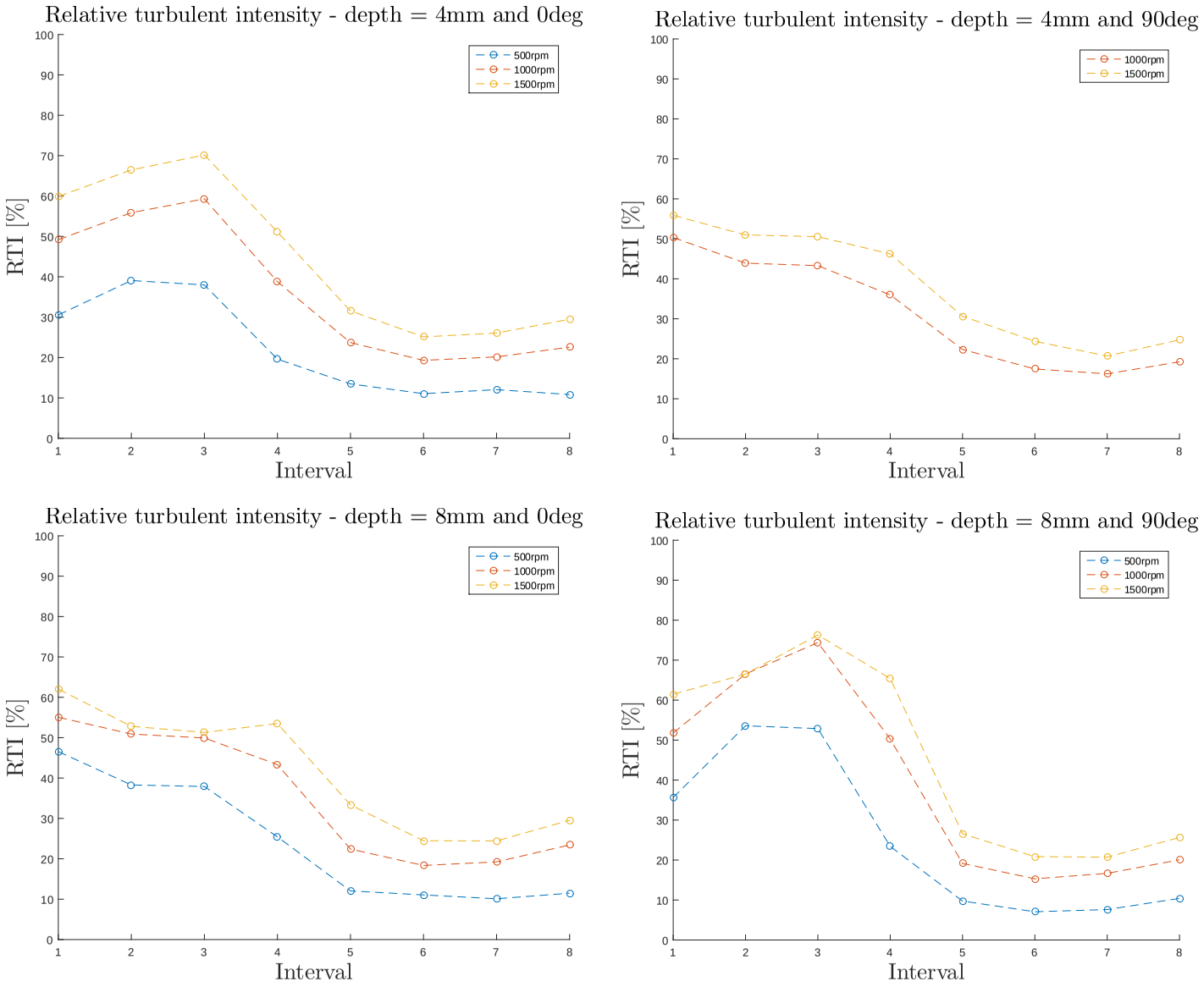


Figure 58: Relative turbulent intensity with varying engine speed

It can be seen, that the RTI tends to follow the shape of the curve from the average velocities. The RTI is higher during the compression stroke than the expansion stroke, with its maximum occurring around the second or third interval. The curves starts out different at BDC, but all have the decreasing form down to TDC, and are then again more constant, with a small tendency of rising close the the second BDC. It can be expected to see a lower RTI at higher pressure, because the high pressure can limit the movement of the eddies, and thereby lowering the RTI.
It can seen, that the RTI also is increased when the engine speed goes up. No big and consistent changes occurs by changing the position of the hot-wire.

From the average velocity and the RTI, it seems like the hot-wire is capturing the changes in the engine regarding crank angle position, variation of engine speed and position of the measurement.

6.3 Autocorrelation

In order to find the integral scales, the autocorrelation function have been calculated. An autocorrelation function have been calculated for each interval for the 12 measurements, all these can be seen in Appendix K. The autocorrelation is calculated from the fluctuations. But because of the shape of the velocity signal, finding and subtracting the average velocity from the signal is not adequate. A high-pass filter of $200Hz$ was used to separate the fluctuations from the velocity signal. With the fluctuations isolated, a representative measurement have been selected and can be seen in Figure 59.

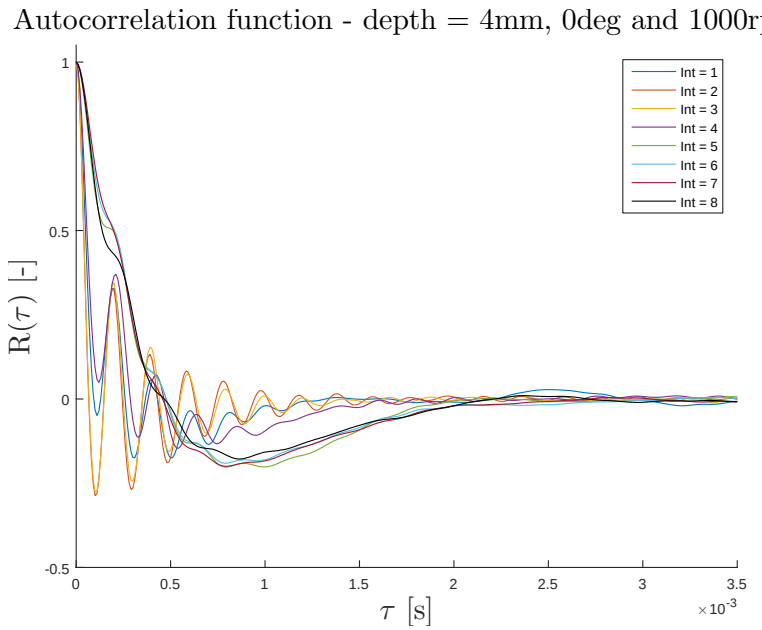


Figure 59: Representative autocorrelation function. Position at $4mm$, $0deg$ and $1000rpm$

From the figure it can be seen that there are two main behaviours of the autocorrelation functions. One behaviour is the one which look like the autocorrelation function form the pipe test. Starts around one and smoothly decrease and stabilize at zero. This

is the case for intervals 5-8. These intervals represents the expansion stroke, and from the average velocities and RTI it can be seen that the velocity is low with a low RTI. This means that the flow is more steady and there is a bigger correlation between the points in time. For the intervals 1-4 the autocorrelation is alternating before stabilizing. These intervals represent the compression stroke, and here the average velocity and RTI is much higher. The RTI is up to 75%, meaning that there is a high level of turbulence compared to the mean flow. The high level of turbulence indicate a flow which is highly discorded, making a correlation between points in time very weak. This can be seen as the alternating autocorrelation functions. The tendency of having alternating autocorrelation functions during the compression stroke and more smooth functions during the expansion stroke, can be seen in all the measurements. By changing the engine speed, the slope of the autocorrelation function differs, and are getting steeper as the engine speed goes up. This can be explained by the higher RTI, which can make the flow more chaotic and thereby lead to a faster loss of cohering in the flow.

The integral time scale for each interval and for all measurement have been found from the autocorrelation function. It was found as the first zero crossing as in the pipe test. The integral length scale was calculated from the integral time scale, and the results can be seen in fig. 60.

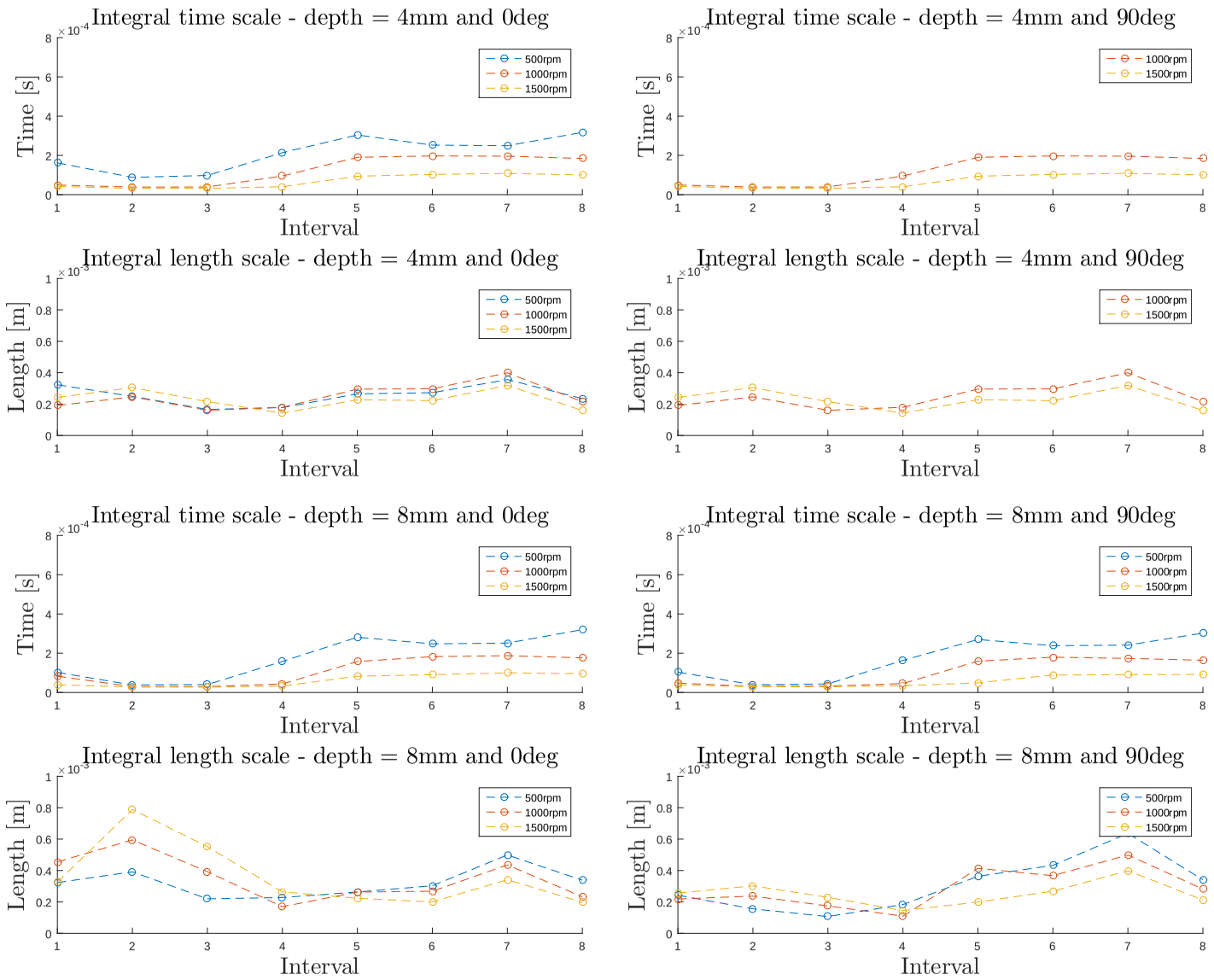


Figure 60: Integral scales with varying engine speed

Looking at the integral time scales, it can be seen that time scale is low during the compression stroke. Because the time scale is found from the autocorrelation function, and that the autocorrelation function is alternating here, a very low time scale can be expected. This means that the results of the time scale during the compression stroke is misleading. A better result might be obtained by making a curve fitting, and calculate the time scale from that.
During the expansion stroke, the autocorrelation function is leading to more correct results. It can be seen, that the time scale is higher for the $8mm$ position, which can be expected because there is more space for the eddies to evolve, than for the $4mm$ position. It can also be seen, that the time scale is higher at low engine speeds, this can be explained by the fact that at lower engine speed, the eddies have more time to evolve before the conditions are changed. The angle of the wire does not seems to change the results.

The integral length scale is directly coupled with the integral time scale via the average velocity. It can be seen that the length scale during the compression stroke is low, except for the $8mm$, $0deg$ position. Here the length scale is bigger compared to the other positions, which can be explained by the high average velocity found here.
It can be seen that the length scales a bigger at low engine speed, which is due to the extra time that the eddies have, before the conditions changes.

With the integral time scale found, a check for windowing is performed. It can be seen that the time scale is around $0.005s$. From Section 2.4.2, it was found that the block time should be 400 time the integral time scale, leading to a block time of $2s$. At $1500rpm$ the block time - time of one interval - is $0.005s$, meaning that the biggest eddies might not will be taking into account and windowing can contaminate the energy spectrum, as described in Section 6.1.3.

The Taylor microscale have been calculated for each interval and for every measurement, and can be seen in Figure 61.

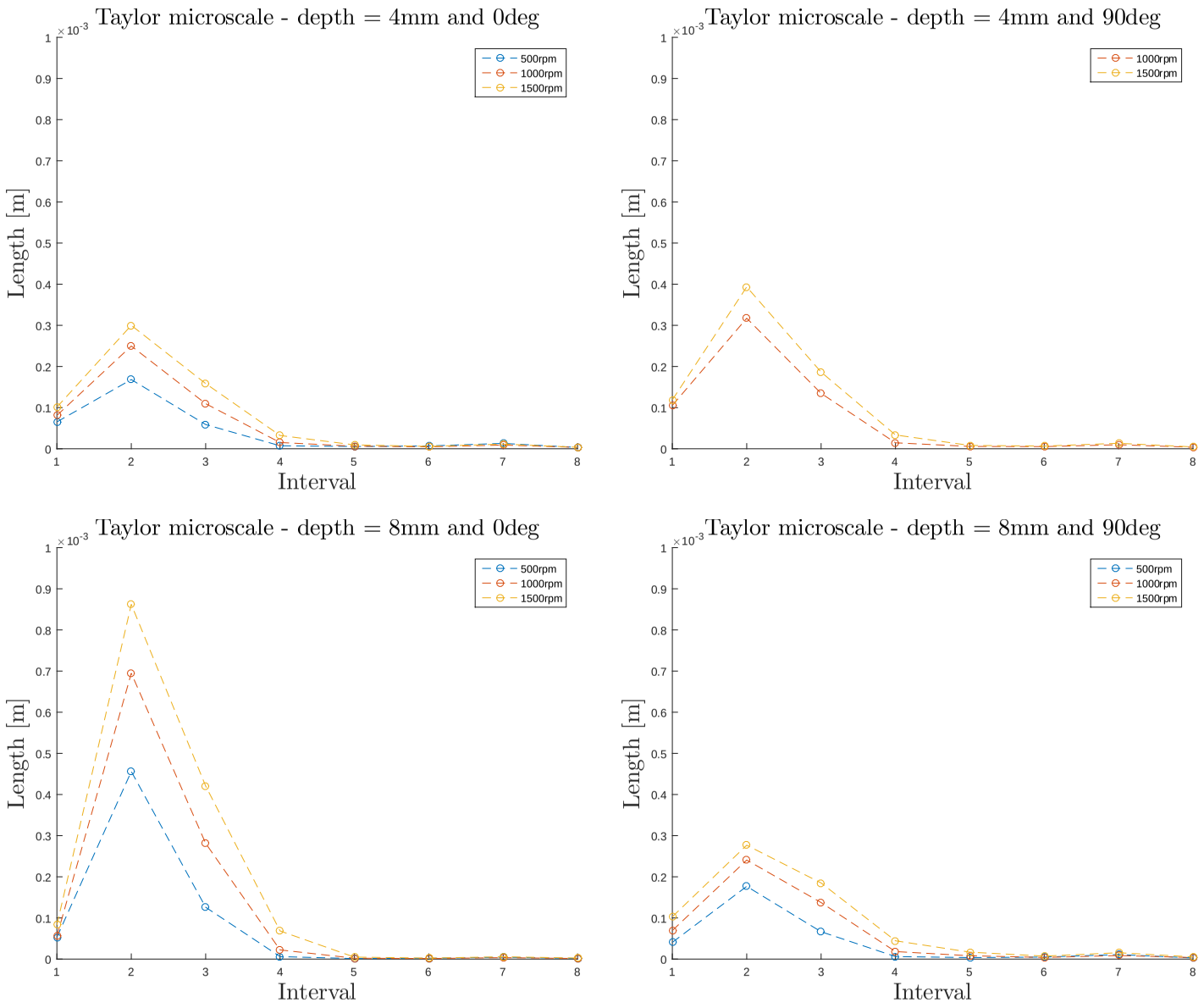


Figure 61: Taylor microscales with varying engine speed

From the figure it can be seen that the Taylor microscales is higher during the compression stroke than during the expansion stroke. From Equation (40) it can be seen that the average velocity affect the Taylor microscale, and the highest average velocity can also be found during the compression stroke. Which is also why the miscroscale at the $8mm$, $0deg$ position are much higher than for the other positions. A higher Taylor microscale also indicates that the transition from the inertial subrange to the dissipation subrange is appearing at larger eddies sizes.
During the expansion stroke, the Taylor microscale seems more constant, which is the same tendency seen from the average velocity and RTI.
It can be seen that the Taylor microscale is higher at higher engine speeds, again meaning that the transition from the inertial subrange to the dissipation subrange is appearing at larger eddies sizes.

The integral times scale and the Taylor microscales have been plotted together with the representative autocorrelation function from before, in order to verify the results, and can be seen in Figure 62. The plot have zoomed in on the start of the curves where the integral time scale and Taylor microscale can be expected to be found.

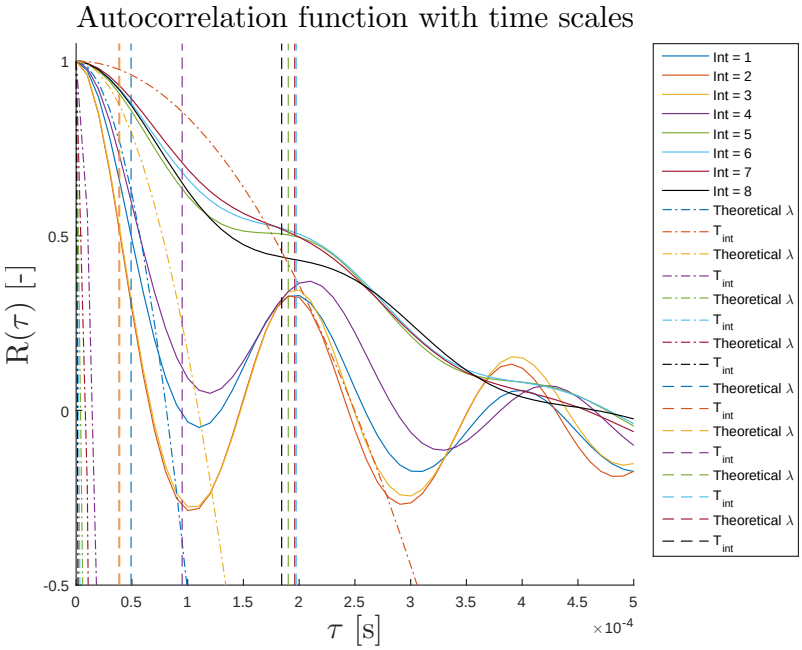


Figure 62: Representative autocorrelation function with T_{int} and λ. Position at $4mm$, $0deg$ and $1000rpm$

The integral time scale at interval 2 and 3 are very similar and therefore look like one line the plot, and the same goes for the integral time scale at interval 6 and 7. Looking at the integral time scales for interval 5 to 8, it can be seen that they represent the area under the curve well. For the interval 1 to 4, it seems like the integral time scale is to small, which again can be explained by the alternating curve.
Looking at the theoretically found Taylor microscale, it can be seen that only the curve fitting from interval 1 to 3 is seems to follow the autocorrelation close to $\tau = 0$. The curve fitting from interval 4 to 8 all drop very rapidity and does not follow the autocorrelation.

But looking at the zero crossing of the curve fittings, it can be seen that they match the value of the Taylor micro scale found in Figure 61.

6.4 Energy spectrum

The energy spectrum for each interval and for all measurement have been calculated, by using MatLab's FFT function multiplies with $dt = \frac{1}{SR} = \frac{1}{100000} = 10^{-5}s$. The energy spectra can be seen in Appendix L. A representative spectrum can be seen in Figure 63.

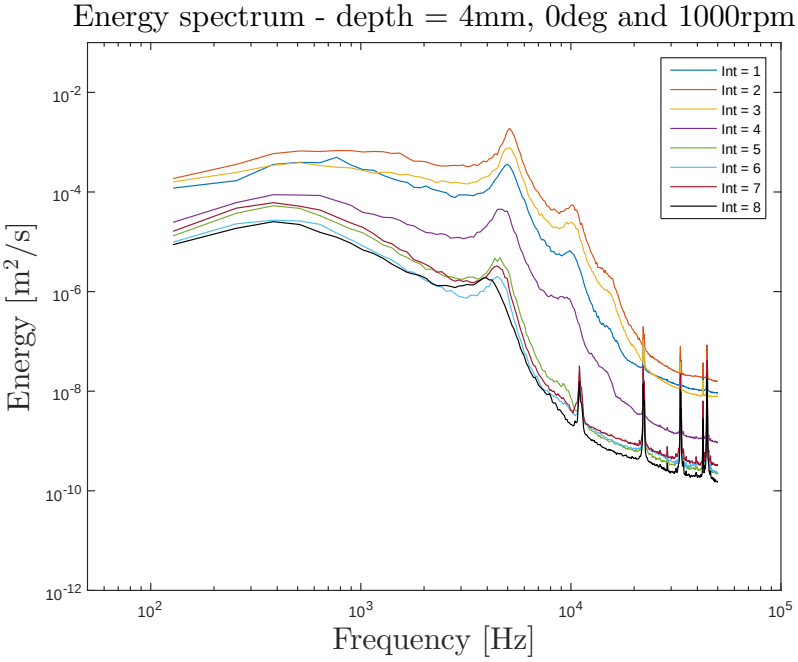


Figure 63: Representative energy spectrum. Position at $4mm$, $0deg$ and $1000rpm$

The area under the plots have been calculated and can be seen in Table 12 compared to the variation in order to validate the energy spectrum.

Interval	V_{var}	**Area**
1	4.646	3.740
2	16.219	12.960
3	7.600	6.078
4	0.705	0.567
5	0.184	0.150
6	0.104	0.087
7	0.211	0.175
8	0.090	0.073

Table 12: Comparison of area under the energy spectra and variance of velocity signal at $4mm$, $0deg$ and 1000

It can be seen that the area under the curves are higher than the variance of the velocity signal, which can be explained by the filtering of the signal, meaning that some

of the energy have been taken out of the signal.
Looking at the first part of the energy spectra it can be seen that a flat plateau obtained in the pipe test is not as pronounced here. Keep in mind, that the spectra are conducted for $45CAD$ intervals, they might have to be raised and starting earlier. It can be seen that the curves for intervals with higher RTI is at higher energy levels, due to the grater amount of turbulent energy compared to the main flow.
In order to understand the first part of the spectrum better, the integral length scale have been converted to a frequency and plotted together with the energy spectrum, this can be seen in fig. 64.

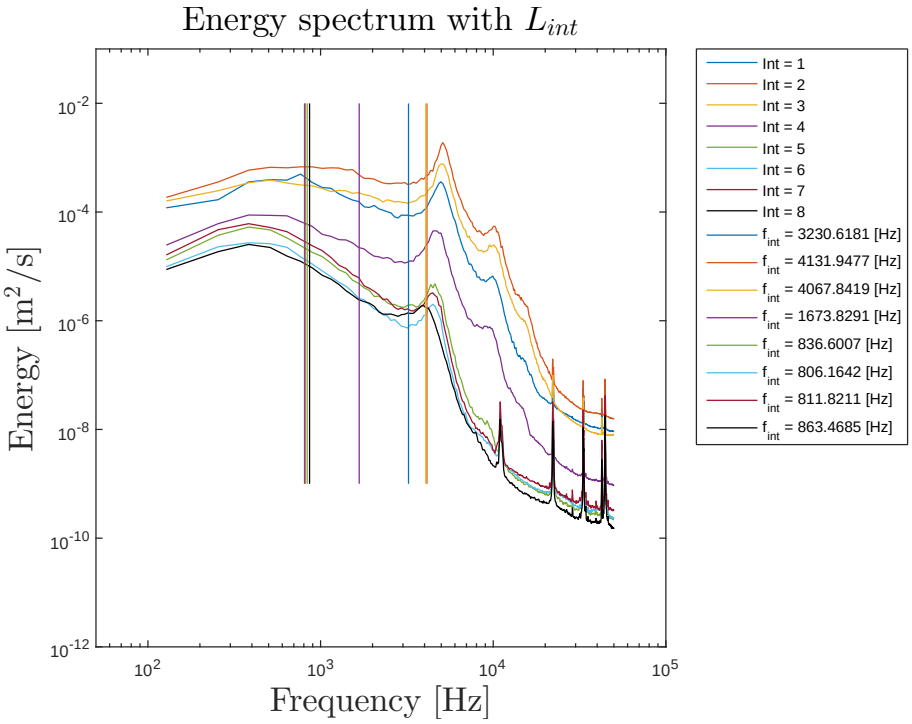


Figure 64: Representative energy spectrum with L_{int} expresed as f_{int}. Position at $4mm$, $0deg$ and $1000rpm$

From the figure it can be seen that the even though the integral length scales from Figure 60 seems similar across the intervals, they differ when transformed into frequencies due to the average velocity. It can also be seen, that the integral length scales are so small, that the low-pass filter of $10kHz$ seems to be to small. The very small eddie sizes can also be a consequence of oscillation the autocorrelation function. It can be seen that the integral length scales in interval 4-8 does not have the flat plateau which was expected, but enters the inertial subrange faster than the for interval 1-3. A peak can be seen at frequency around $4500Hz$, which might come from noise.

Entering the inertial subrange, the $\frac{-3}{5}$ slope have been plotted in order to test the $\frac{-3}{5}$ rule. This can be seen in Figure 65.

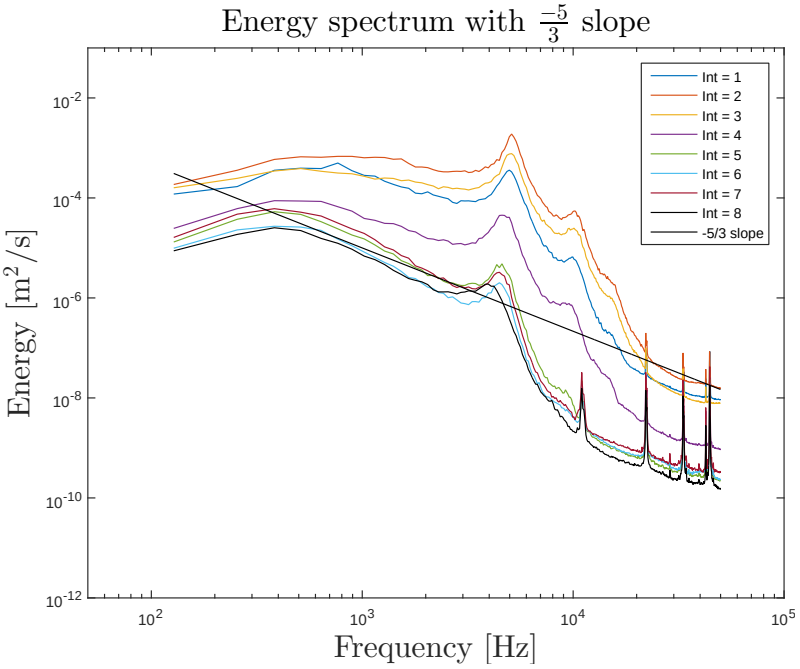


Figure 65: Representative energy spectrum with the $\frac{-3}{5}$ slope. Position at $4mm$, $0deg$ and $1000rpm$

It can be seen that the energy spectra from interval 4 to 8 follows the slope fine after the integral length scale. The spectra from interval 2 and 3 does not seem to follow the slope, which make it looks like that there is an error connected to these.
In order to find the boundaries of the inertial subregion the Taylor microscale have been plotted together with the energy spectra and can be seen in Figure 66.
Here it can be seen that the Taylor microscales seems to be bigger than the integral time scale in interval 1 to 3. This does not make physical sense, but can be used as a measure of the oscillating autocorrelation function and be an indication of a very short inertial subrange for these intervals. For the interval 4-8 it can be seen that the inertial subrange ends after the low-pass filer, which indicated that the low-pass filter were set to low to fully show the spectrum.

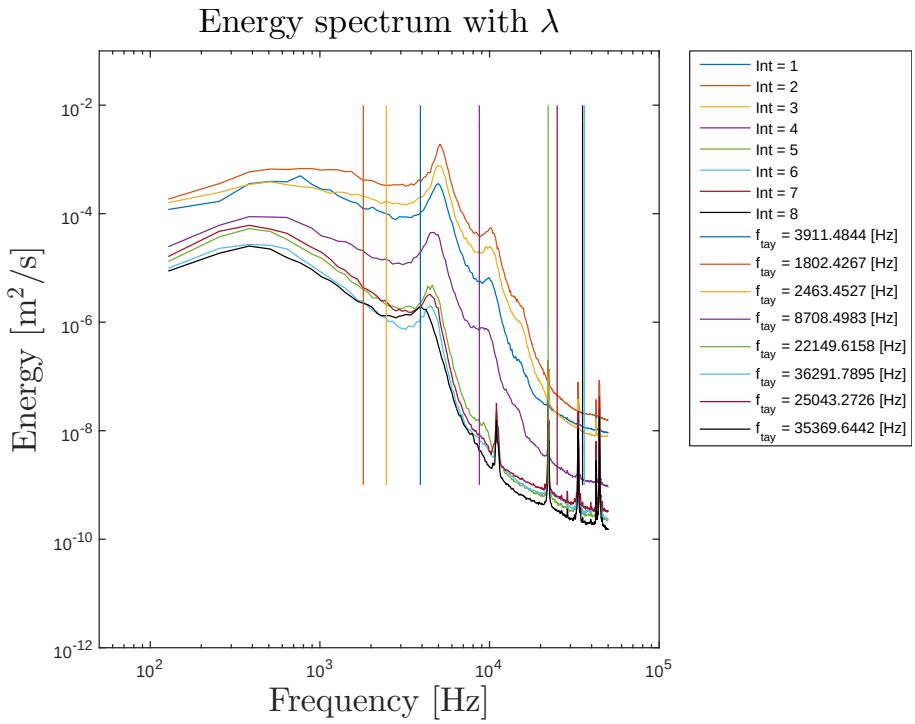


Figure 66: Representative energy spectrum. Position at $4mm$, $0deg$ and $1000rpm$

In order to find the smallest eddies, the Kolmogorov scales was calculated. The Kolmogorov scales was calculated from the dissipation rate and the viscosity found from at the mean gas temperature and pressure. The results from the representative $4mm$, $0deg$ and $1000rpm$ position and can be found in Table 13.

Interval	$\boldsymbol{\varepsilon \cdot 10^5 [\frac{m^2}{s^3}]}$	$\boldsymbol{\nu \cdot 10^{-6} [\frac{m^2}{s}]}$	$\boldsymbol{T_{kol} \cdot 10^{-5}}$ **[s]**	$\boldsymbol{L_{kol} \cdot 10^{-5}}$ **[m]**
1	0.815	9.817	1.097	1.038
2	0.301	9.817	1.807	1.332
3	0.727	9.817	1.162	1.068
4	3.435	9.817	0.535	0.724
5	5.087	9.817	0.439	0.657
6	5.404	9.817	0.426	0.647
7	2.960	9.817	0.576	0.752
8	8.957	9.817	0.331	0.570

Table 13: Kolmogorov time and length scales

Here it can be seen that the dissipation rate is much higher than for the pipe test. This means that the dissipation happens faster, which also can be expected due to the behaviour of the engine. All Kolmogorov scales have been plotted and can be seen in Figure 67.

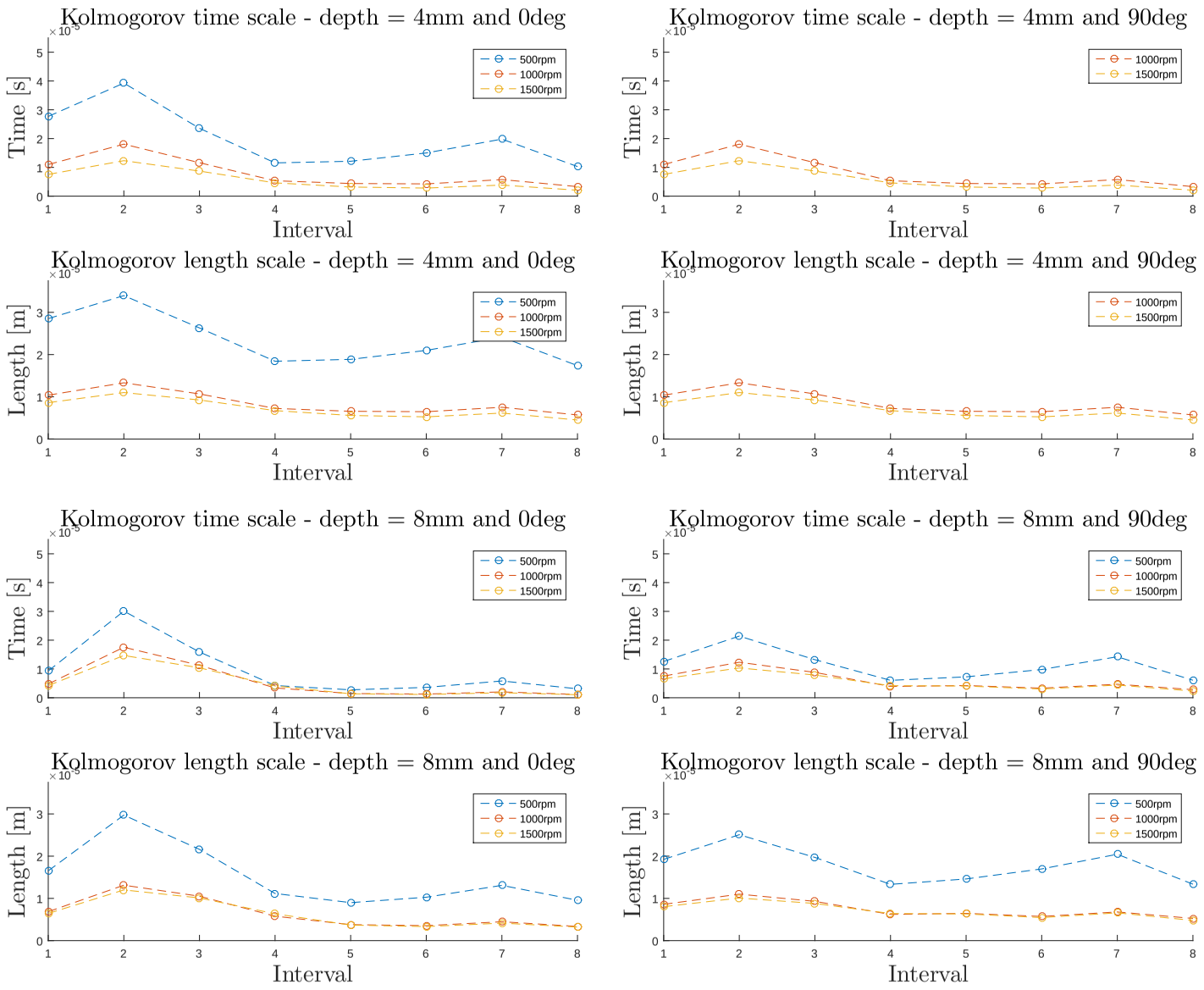


Figure 67: Kolmogorov scales with varying engine speed

Looking at the kolmogorov time scale, it can be seen that there is a tendency for the time scale to rise and fall during both the compression and expansion stroke. But the highest time scales seems to be found during the compression stroke. The highest times scales can be found at low engine speeds, again due to the extra time during these revolutions.
The Kolmogorov length scales are directly linked to the time scale via the average velocity, and is thereby following the behaviour of the time corresponding time scale.

With the Kolmogorov length scale found, it can be plotted together with the energy spectra. This can be seen in Figure 68.

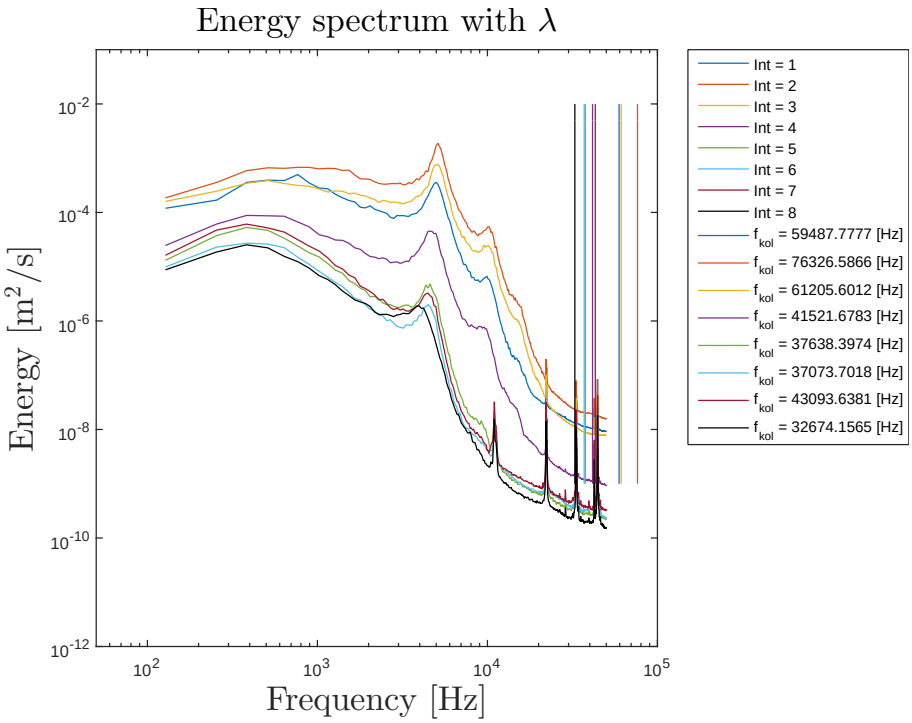


Figure 68: Representative energy spectrum. Position at $4mm$, $0deg$ and $1000rpm$

From the figure it can be seen, that the kolmogorov length scales are so small, that they are outside the measuring range, even with a sample rate of $100kHz$. The energy spectra for interval 4 to 8 seems to be correct, but the energy spectra for interval 1 to 3 seems to be too turbulent, that an analysis with the relative simple experimental set-up used in this study is not adequate.

7 Discussion

The purpose of this study was to obtain knowledge and experience regarding the use of the Hot-Wire Anemometry technique, and stablish a basis for turbulent measurements in IC engines.

A first experience of the Hot-Wire Anemometry technique was obtained via the pipe test. A pipe flow is a well known flow, and a good way to get a fist feeling of the hot-wire. The test set-up used to perform the pipe test was a simple set-up which had some flaws. The holes drilled in order to insert the hot-wire had a big effect on the flow, which was seen from the velocity profile. The holes which was not used was covered with tape in order to seal the pipe. To optimize the test set-up a way to seal the used hole has to be developed. Also effect from the pipe inlet was found to disturb the flow. The pipe flow was very sensitive to movement in front of the inlet, and a solution of mounting a filter and tubes decreased the sensitivity. But effect of the inlet was still found. In order to optimize the set-up a prober designed nozzle could be mounted. This would improve the stability of the flow and the boundary layer. The fixture made to easy adjust the hight of the hot-wire could be improved, such that a more accurate hight could obtained. But for these experiments, it was found to be adequate.
Despite the influence from the holes and inlet, good measurement was able to be made. The results could be compared with theoretically obtained results and made sense from a physical point of view. It was found that the low-pass filter in the miniCTA was to low. The Kolmogorov length scales was outside the low-pass filters range. In order to make a prober filtration of the signal, a devise with a higher low-pass filter setting has to be used. In order to separate the fluctuation from the velocity signal, a high-pass filter could also be used in the pipe test. This could lead to a better separation and filtering of flows with other directions than intended as seen in Section 6.1.1.
The pipe test gave a good understanding of equipment and the theory behind the use of a hot-wire. This experience was used in the design of the engine test.

The engine test performed in this study is used to establish a basis for turbulent measurement for further studies. In order to get a first impression of engine measurements, simplification was made to establish a measurement procedure.
The flow in an engine is highly three dimensional, so by using a 1D probe, the signal can be contaminated from flow in other directions than the intended. This means that the velocity obtained can appear higher that it actually is. In order to properly resolve the flow in the three dimensions, a 3D high-temperature probe has to be used. In order to use the 3D probe correctly, a more advanced calibration procedure has to be used. Another simplification was used, due to a simple simulation for calculating the gas temperature and pressure, in order to perform a temperature correction. A more advanced simulation which for example takes heat loss to the cylinder into account can be used, in order to obtain a more accurate estimation of the temperature and pressure. An other solution could be to use a hot-wire probe with an integrated thermocouple, which directly adjust for elevated temperatures.
In order to perform the temperature correction, a simple equation which only takes convection into account. A more acculturate heat transfer equation could be obtained by taking conduction to the prongs into account. And last, a mean viscosity was used for calculating the Kolmogorov scales. Because the engine cycle was divided into intervals a

viscosity for each interval could be found.

The fixture designed for the engine test was tested and worked as intended. The Swagelok fitting was able to hold the wire support in place and together with the adaptor able to mount the hot-wire in the engine. For further testing, a solution where the hot-wire does not have to be put through the fixture and where the position of the hot-wire was easier to adjust would be preferred. A probe support with integrated thread can be produced by DanTec. This will solve the mounting problem, but the adjustment problem would still be present.
During the data processing, it was discovered that problems with the inductive sensor had occurred. An interpolation was made in order to manually add a point under the assumption of constant engine speed. The inductive sensor was shifted from the TDC of the engine, which means that an estimation of the shift had to be made, and accuracy of the position of the piston was lost. A solution which could improve the precision of the position of the piston could be to use an encoder instead.

The miniCTA used in the engine test was the same type as the one for he pipe test. This means that the highest low-pass filter was $10kHz$, which was not enough for the pipe test. So it was expected that it was not sufficient for the engine test either, which was also the case. But it was decided that it was sufficient for this test. The miniCTA had the option of using a higher overheat ration than the 0.8 used. 0.8 was used in order to protect the hot-wire, and with a wire temperature of 520 $°C$ and a flow of just over 300 $°C$ it was assumed to be adequate. For further tests, the overheat ratio can be raised.

A problem regarding windowing was discovered. The problem occurs because of short interval time. This is a problem which is hard to avoid. Even at relative low engine speeds, the engine is still spinning to fast. And by lowering the amount of intervals, the period where the signal is observed is getting to big to make prober sense. An improvement to this could be to make use of an interval 400 times the integral time scale (for $1000rpm$ this was found to be $120CAD$). But in stead of having the intervals after each other, shifting the interval. The interval could be moved a few CAD at the time in order to resolve the entire cycle.

A challenge regarding the separation of the fluctuations from the velocity signal was found. It was found that a high-pass filter was able to make a good separation, but has to be set-up such that important information is not lost. In this study a fixed high-pass filter was used, but in order to obtain better results, a high-pass filter have to be tailored to each measurement.

A problem regarding the autocorrelation function was encountered. The flow seems to be too turbulent for the autocorrelation function to make a prober correlation, which leads to incorrect results of the integral time and length scale. A higher sampling rate could help the problem, because the time step, τ, will be reduced and the autocorrelation function will thereby have a more detailed picture of the flow.

For future work it seems like an upgrade of equipment is necessary in order to obtain prober results. The system has to be able to have a higher sample rate and a higher low-pass filter. This can be seen from Figure 67 because the smallest eddies in the flow is

outside the measurable area. Further more, a 2D or 3D wire would improve the accuracy of the velocity measurement due to a better resolving of the flow directions. An encoder would help determine the position of the piston better. An investigation of the influence of a higher overheat ratio could be carried out in further work.
It has been found that a good data processing is important for obtaining good results. By using appropriate filtering and using a new way to implement the intervals, better results of the time and length scales can be obtained.

8 Conclusion

The purpose of this study was to obtain knowledge and experience in the use of hot-wire anemometry, and establish a basis for measurement of turbulent structures in an IC engine. In order to obtain a first experience of hot-wire anemometry a simple pipe experiment have been conducted.
From the pipe test an experience in the use of the hardware, testing procedure and data processing was achieved. The different elements of the measuring chain were examined and the function of each element was learned. It was found that the pipe test set-up had flaws regarding the holes for inserting the hot-wire and in inlet. But the hot-wire system was able to recognise the effects that these flaws created and recreate known results for a pipe flow. A limitation of the system was the low-pass filter, which filtered eddy sizes equal to the Kolmogorov length scale out .

With the knowledge from the pipe test it was possible to set-up and conduct measurement for the engine test. It can be concluded that the engine set-up was able to produce a volt signal with a corresponding pulse from the inductive sensor to determine the position of the piston.
From the test of the fixture, which was developed to install the hot-wire in the engine, it can be concluded that the fixture can sufficiently hold on to the hot-wire support without damaging it. The elements in the measuring chain in the engine set-up were investigated and it could be concluded that the high-temperature probe and the support could withstand the high temperature and pressure in the engine. A 3D high-temperature probe would be preferred in order to resolve the flow better. It was found that the miniCTA used did not have the needed settings in terms of low-pass filter which was needed to include all eddy sizes that the hot-wire could resolve. Hardware which have a higher low-pass filter than $10kHz$ is needed in future measurement for including the Kolmogorov scales. The A/D module connected to the miniCTA had a maximum sampling rate of $100kHz$, but it was shown that the Kolmogrov scales where found at higher frequencies. So a module with a higher sampling rate than $100kHz$ would be preferred.

From this study it can be concluded that data processing is an important element of conducting turbulent scales. The first step of the data processing is temperature correcting, which was found to have significant effect and has to be performed. The estimated temperature and pressure used in the temperature correction can be estimated from a simple simulation and give good results. To improve the temperture correcting, a more advanced simulation has to be performed or a hot-wire with integrated thermocouple can be used.
It was found that the separation of fluctuations from the velocity signal was best performed with a high-pass filter. And it was found, that the length of the interval which the velocity signal was divided into, has a significant effect on the turbulent scales. A prober method for dividing the velocity signal into intervals has to be developed in order to obtain prober results of the turbulent scales.

It can be concluded that turbulent structures can be obtained by hot-wire anemometry. A basis for measuring turbulent structures in an IC engine have been established during this study. With the basis, further investigation can be made for obtaining more accurate results, which can be used to understand the combustion process in engines further.

Bibliography

[1] Dantec Dynamic, Technical Data For Miniature Wire Sensors, 20/10-2015, URL: *http://www.dantecdynamics.com/products-and-services/single-sensor-miniature-wire-probes*

[2] D. R. Lancaster, *Effects of turbulence on spark-ignition engine combustion*, PhD thesis, University of Illinois, 1975

[3] Gíslason, Kjartan Steiner and Lloran, Eduard Gómez, *Design of an experimental SI engine setup with water injection to enable enhancement of compression ratio*, Master thesis, DTU, June 2015

[4] Hari, Kristof, *Measurement of gas mixing in a pipe flow by the use of a tracer gas method*, 23/07-2015

[5] H. H. Bruun, *Hot-Wire Anemometry - Principles and Signal Analysis*, Oxford university press, 1995

[6] H. Tennekes and J. L. Lumley, *A First Course in Turbulence*, The MIT Press, 1974

[7] Jørgensen, Finn. E, *How to measure turbulence with hot-wire anemometers - a practical guide*, 2002

[8] Kenneth K. Kuo and Ragini Acharya, *Fundamentals of turbulent and multiphase combustion*, Wiley, 2012

[9] L. W. B., Browne & A, Dinkelacker, *Turbulent pipe flow: pressures and velocities*, Fluid Dynamics Research, 1995

[10] Pijush K. Kundu, Ira M. Cohen og David R. Dowling, *Fluid Mechanics*, 5. edition, Elsevier Science and Technology Books

[11] P. L. O'Niell, D. Nicolaides, D. Honnery and J. Soria, *Autocorrelation Functions and the Determination of Integral Length with Reference to Experimental and Numerical Data*, 15th Australian Fluid Mechanics Conference The University of Sydney, Sydney, Australia. 2004.

[12] P. O. Witze, *A critical comparison of hot-wire anemometry and laser doppler velocimetry for I.C. engine applications*, SAE Technical Paper no. 800132, 1980

[13] Spencer C. Sorensen, *Engine Principles and Vehicles*, Department of Mechanical Engineering DTU, 13th August 2012

[14] S.Subramaniyam, V. Ganesan, P. Srinivasa Rao, *Turbulent flow inside the cylinder of a diesel engine - an experimental investigation using hot wire anemometer*, Springer-Verlag, 1990

[15] Velte, Clara M., Slideshow: *Notes for EFD course 2013 Jan. 2013 - Signal processing of flow measurements*, Jan. 2013

[16] WaveMetrics, 1/12-2015, URL: *https://www.wavemetrics.com/products/ igorpro/-dataanalysis/signalprocessing/spectralwindowing.htm*

Appendices

Appendix A

Excel sheet for pressure - velocity conversion

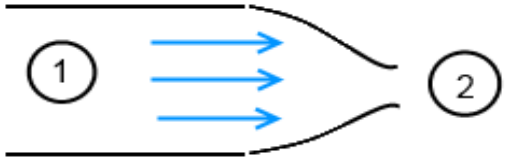
Constants:		
d1 =	0.24	[m]
d2 =	0.1	[m]
T (atm) =	23.4	[C]
p2 (atm) =	99232	[kg/(s^2*m)]
rho (from EES) =	1.185	[kg/m^3]
1/2*(1-(d2/d1)^4) =	0.484929591	[-]

From p -> v		
Δp =	20	[kg/(s^2*m)]
p1 =	99252	[kg/(s^2*m)]
v2 =	5.899517308	[m/s]

From v -> p		
v2 =	7	[m/s]
p1 =	99260.15744	[kg/(s^2*m)]
Δp =	28.1574367	[kg/(s^2*m)]

Governing equ:

$$\frac{(p_1 - p_2)}{\rho} = \frac{1}{2} v_2^2 \left(1 - \left(\frac{d2}{d1}\right)^4\right)$$



Appendix B

Excel file with data and file names

The Excel sheet presented here is the one used, when the experiments was carried out. The Excel sheet was afterwords broken up into seven sheets, when the Matlab script had to read from it. The sheets created was one for the velocity profiles and one for each of the scripts for the data bases.

Genneral comments
When book and high_fq experiments was performed, the time between the blocks was 1 sec

Date:	12-11-2015

# Test run	X-position *D [m]	Y-position *D [m]	linial	Temp. (atm) [C]	Press. (atm) [Pa]	Orifice ΔP [Pa]	Sampling rate [Hz]	# Samples	# Blocks	Calibration file	Filename	Comment
1	-5	0.500	17.300	23.0	101150	345	40000	96000000	400	final_cal	book_m5_0500	2 walked by
2	-5	0.675	20.800	23.9	101204	345	40000	96000000	400	final_cal	book_m5_0675	6 walked by
3	-5	0.850	24.300	24	101284	345	40000	96000000	400	final_cal	book_m5_0850	4 walked by
4	-5	0.000	7.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0000	
5	-5	0.025	7.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0025	
6	-5	0.050	8.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0050	
7	-5	0.075	8.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0075	
8	-5	0.100	9.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0100	
9	-5	0.125	9.800	23.7	101375	345	50	50	1	final_cal	vel_m5_0125	
10	-5	0.150	10.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0150	
11	-5	0.175	10.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0175	
12	-5	0.200	11.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0200	
13	-5	0.225	11.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0225	
14	-5	0.250	12.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0250	
15	-5	0.275	12.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0275	
16	-5	0.300	13.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0300	
17	-5	0.325	13.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0325	
18	-5	0.350	14.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0350	
19	-5	0.375	14.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0375	
20	-5	0.400	15.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0400	
21	-5	0.425	15.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0425	
22	-5	0.450	16.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0450	
23	-5	0.475	16.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0475	
24	-5	0.500	17.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0500	
25	-5	0.525	17.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0525	
26	-5	0.550	18.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0550	
27	-5	0.575	18.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0575	
28	-5	0.600	19.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0600	
29	-5	0.625	19.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0625	
30	-5	0.650	20.300	23.8	101375	345	50	50	1	final_cal	vel_m5_0650	
31	-5	0.675	20.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0675	
32	-5	0.700	21.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0700	
33	-5	0.725	21.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0725	
34	-5	0.750	22.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0750	
35	-5	0.775	22.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0775	
36	-5	0.800	23.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0800	
37	-5	0.825	23.800	23.8	101375	345	50	50	1	final_cal	vel_m5_0825	
38	-5	0.850	24.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0850	
39	-5	0.875	24.800	23.9	101375	345	50	50	1	final_cal	vel_m5_0875	
40	-5	0.900	25.300	23.9	101375	345	50	50	1	final_cal	vel_m5_0900	
41	10	0.850	24.300	23.7	101443	345	40000	96000000	400	final_cal	book_m5_0850	3 walked by
42	10	0.000	7.300	23.7	101461	345	50	50	1	final_cal	vel_10_0000	
43	10	0.025	7.800	23.7	101461	345	50	50	1	final_cal	vel_10_0025	
44	10	0.050	8.300	23.8	101461	345	50	50	1	final_cal	vel_10_0050	
45	10	0.075	8.800	23.9	101461	345	50	50	1	final_cal	vel_10_0075	
46	10	0.100	9.300	23.9	101461	345	50	50	1	final_cal	vel_10_0100	
47	10	0.125	9.800	23.9	101461	345	50	50	1	final_cal	vel_10_0125	
48	10	0.150	10.300	23.9	101461	345	50	50	1	final_cal	vel_10_0150	
49	10	0.175	10.800	23.8	101461	345	50	50	1	final_cal	vel_10_0175	

50	10	0.200	11.300	23.8	101461	345	50	50	1	final_cal	vel_10_0200	
51	10	0.225	11.800	23.9	101461	345	50	50	1	final_cal	vel_10_0225	
52	10	0.250	12.300	23.8	101461	345	50	50	1	final_cal	vel_10_0250	
53	10	0.275	12.800	23.8	101461	345	50	50	1	final_cal	vel_10_0275	
54	10	0.300	13.300	23.8	101461	345	50	50	1	final_cal	vel_10_0300	
55	10	0.325	13.800	23.8	101461	345	50	50	1	final_cal	vel_10_0325	
56	10	0.350	14.300	23.8	101461	345	50	50	1	final_cal	vel_10_0350	
57	10	0.375	14.800	23.8	101461	345	50	50	1	final_cal	vel_10_0375	
58	10	0.400	15.300	23.9	101461	345	50	50	1	final_cal	vel_10_0400	
59	10	0.425	15.800	23.8	101461	345	50	50	1	final_cal	vel_10_0425	
60	10	0.450	16.300	23.8	101461	345	50	50	1	final_cal	vel_10_0450	
61	10	0.475	16.800	23.9	101461	345	50	50	1	final_cal	vel_10_0475	
62	10	0.500	17.300	23.9	101461	345	50	50	1	final_cal	vel_10_0500	
63	10	0.525	17.800	23.9	101461	345	50	50	1	final_cal	vel_10_0525	
64	10	0.550	18.300	23.9	101461	345	50	50	1	final_cal	vel_10_0550	
65	10	0.575	18.800	23.9	101461	345	50	50	1	final_cal	vel_10_0575	
66	10	0.600	19.300	23.8	101461	345	50	50	1	final_cal	vel_10_0600	
67	10	0.625	19.800	23.8	101461	345	50	50	1	final_cal	vel_10_0625	
68	10	0.650	20.300	23.8	101461	345	50	50	1	final_cal	vel_10_0650	
69	10	0.675	20.800	23.9	101461	345	50	50	1	final_cal	vel_10_0675	
70	10	0.700	21.300	23.8	101461	345	50	50	1	final_cal	vel_10_0700	
71	10	0.725	21.800	23.8	101461	345	50	50	1	final_cal	vel_10_0725	
72	10	0.750	22.300	23.7	101461	345	50	50	1	final_cal	vel_10_0750	
73	10	0.775	22.800	23.8	101461	345	50	50	1	final_cal	vel_10_0775	
74	10	0.800	23.300	23.8	101461	345	50	50	1	final_cal	vel_10_0800	
75	10	0.825	23.800	23.9	101461	345	50	50	1	final_cal	vel_10_0825	
76	10	0.850	24.300	23.8	101461	345	50	50	1	final_cal	vel_10_0850	
77	10	0.875	24.800	23.8	101461	345	50	50	1	final_cal	vel_10_0875	
78	10	0.900	25.300	23.8	101461	345	50	50	1	final_cal	vel_10_0900	
79	10	0.675	20.800	23.7	101462	345	40000	96000000	400	final_cal	book_10_0675	1 walked by
80	10	0.500	17.300	23.7	101470	345	40000	96000000	400	final_cal	book_10_0500	

Appendix C

MatLab script for velocity profile measurements

The main script is here presented.

```
clear all
close all
clc

BuildRawDataBase = 0;
LoadRawDataBase = 1;
RawDataBaseFileName = 'dbraw';

BuildWorkDataBase = 0;
LoadWorkDataBase = 1;
WorkDataBaseFileName = 'db';

xlsversion = 'xls97'; % Options are 'xls97' and 'xls95'. MAC shoud use...
   'xls95'

%...
  ========================================================================...

% Either loading all data from excelsheets and LabView files into a
% Matlab structure or from a preveously generated Mat file
if BuildRawDataBase
    clear db

    % Properties of the excelsheets that should be stripped
    ExcelProperties = {
    % (path\)filename                          1st header row    2nd ...
      header row
    'final_test_vel_profiles'     'Ark1'              3            ...
         5
    };

    % list of variable names for the general data belonging to the 1st...
       header
    VariableList1 = {
    'date'
    };

    % list of variable names for the test data belonging to the 2nd ...
      header
    VariableList2 = {
    'test_nr','x_pos','y_pos','T_atm','p_atm','orifice_Δ_p',...
    'sampling_rate','nr_samples','nr_blocks','cal_file','file_name','...
      comment'
    };

    % stripping all excel data into a single structure
    dbraw = excelstrip(ExcelProperties,VariableList1,VariableList2,...
      xlsversion);

    % adding time-velocity files to the structure
    dbraw.file_name.val = cell(size(dbraw.file_name.txt));
```

```
43
44      count = 0;
45      for i = 1:length(dbraw.file_name.txt)
46          if ¬isempty(dbraw.file_name.txt{i})
47              dbraw.file_name.val{i} = dlmread([dbraw.file_name.txt{i},'...
                  .txt'],'',126,0);
48              count = count + 1
49          end
50      end
51      save(RawDataBaseFileName,'dbraw');
52      disp('dbraw created')
53  elseif  LoadRawDataBase
54      clear db
55      load(RawDataBaseFileName);
56      disp('dbraw loaded')
57  end
58  %...
    ========================================================================...

59  % Either moving data from raw database into a working database, adding...
       some
60  % general data and doing some general calculations or loading the ...
      working database
61  % from a previously made MAT file
62  if BuildWorkDataBase
63      clear db
64
65      % Genneral data
66      db.totaltest = length(dbraw.test_nr.val);
67      db.nr_samples = dbraw.nr_samples.val(1);
68      db.N_half = floor(db.nr_samples/2);
69      disp('genneral data')
70
71      % Data from dbraw put into db
72      db.test_nr = dbraw.test_nr.val;
73      db.x_pos = dbraw.x_pos.val;
74      db.y_pos = dbraw.y_pos.val;
75      db.T_atm = dbraw.T_atm.val;
76      db.p_atm = dbraw.p_atm.val;
77      db.orifice_Δ_p = dbraw.orifice_Δ_p.val;
78      db.sampling_rate = dbraw.sampling_rate.val;
79
80      db.time = zeros(db.nr_samples,db.totaltest);
81      db.V = zeros(db.nr_samples,db.totaltest);
82      db.dt = zeros(db.totaltest,1);
83      for i = 1:db.totaltest
84          db.time(:,i) = dbraw.file_name.val{i}(:,1);
85          db.V(:,i) = dbraw.file_name.val{i}(:,2);
86          db.dt(i) = db.time(end,i)/db.nr_samples;
87      end
88      disp('dbraw to db')
89
90      % Statistics
91      db.V_avg = zeros(1,db.totaltest);
92      db.fluc = zeros(db.nr_samples,db.totaltest);
93      db.V_rms = zeros(1,db.totaltest);
94      db.var = zeros(1,db.totaltest);
95      db.RTI = zeros(1,db.totaltest);
96      for i = 1:db.totaltest
```

```
97          db.V_avg(i) = sum(db.V(:,i))/db.nr_samples;
98          db.fluc(:,i) = db.V(:,i)-db.V_avg(i);                    % ...
              Flucturations
99          db.V_rms(i) = sqrt(sum(db.fluc(:,i).^2)/db.nr_samples); % ...
              Standart diviation
100         db.var(i) = db.V_rms(i)^2;                               % ...
              Variance
101         db.RTI(i) = db.V_rms(i)./db.V_avg(i);                    % ...
              Relativ turbulence intensity
102     end
103     disp('statistics')
104
105     % Making an index for gartering data-sets with same y position
106     count = 0;
107     for i = 1:db.totaltest
108         if  i == 1
109             count = count + 1;
110             db.y_index(count) = db.y_pos(i);
111         elseif  sum(db.y_index(:) == db.y_pos(i)) == 0
112             count = count + 1;
113             db.y_index(count) = db.y_pos(i);
114         end
115     end
116     disp('y index')
117
118     save(WorkDataBaseFileName,'db','-v7.3');
119     disp('db created')
120 elseif  LoadWorkDataBase
121     load(WorkDataBaseFileName);
122     disp('db loaded')
123 end
124
125 %-------------------------------------------------------------------------
126 %                              PLOTS AND DISPLAIES
127 %-------------------------------------------------------------------------
128 clc;
129 plot_V_avg = 1;
130 plot_RTI = 1;
131
132 index_10 = db.x_pos == 10;
133 index_m5 = db.x_pos == -5;
134 index_bound_10 = db.x_pos == 10 & db.y_pos > 0.15;
135 index_bound_m5 = db.x_pos == -5 & db.y_pos > 0.15;
136
137 my_colormap = [0 0.4470 0.7410; 0.8500 0.3250 0.0980; 0.9290 0.6940 0...
      .1250;...
138                 0.4940 0.1840 0.5560; 0.4660 0.6740 0.1880; 0.3010 0...
                      .7450 0.9330];
139
140 if plot_V_avg
141     V_avg_plots(db,index_10,index_m5,index_bound_10,index_bound_m5,...
          my_colormap)
142 end
143
144 if plot_RTI
145     RTI_plots(db,index_10,index_m5,index_bound_10,index_bound_m5,...
          my_colormap)
146 end
```

Appendix D

MatLab script for turbulent structurs measurements

The script for general calculations and creation of data bases can be seen. This script is the same used to process the data for the 6 positio, but only the first one is presented. For the other position, only the data base name and the name of the Excel sheet has been changed. For this script 1 function for stripping the Excel data sheet can also be seen. The script for collecting the data bases can be found afterwords.

Appendix D.1

Position 10D at 0.500D, db_1

```
clear all
close all
clc

BuildRawDataBase = 0;
LoadRawDataBase = 1;
RawDataBaseFileName = 'dbraw';

BuildWorkDataBase = 1;
LoadWorkDataBase = 0;
WorkDataBaseFileName = 'db_1';

xlsversion = 'xls97'; % Options are 'xls97' and 'xls95'. MAC shoud use...
   'xls95'

%...
  ========================================================================...

% Either loading all data from excelsheets and LabView files into a
% Matlab structure or from a preveously generated Mat file
if BuildRawDataBase
    clear db_1

    % Properties of the excelsheets that should be stripped
    ExcelProperties = {
    % (path\)filename                    1st header row     2nd ...
      header row
    'final_test_energy_1'     'Ark1'              3                5
    };

    % list of variable names for the general data belonging to the 1st...
       header
    VariableList1 = {
    'date'
    };

    % list of variable names for the test data belonging to the 2nd ...
      header
    VariableList2 = {
    'test_nr','x_pos','y_pos','T_atm','p_atm','orifice_Δ_p',...
```

```
35      'sampling_rate','nr_samples','nr_blocks','cal_file','file_name','...
          comment'
36      };
37
38      % stripping all excel data into a single structure
39      dbraw = excelstrip(ExcelProperties,VariableList1,VariableList2,...
          xlsversion);
40
41      % adding time-velocity files to the structure
42      dbraw.file_name.val = cell(size(dbraw.file_name.txt));
43
44      count = 0;
45      for i = 1:length(dbraw.file_name.txt)
46          if ¬isempty(dbraw.file_name.txt{i})
47              block_length = dbraw.nr_samples.val(i)/dbraw.nr_blocks.val...
                  (i)-1;
48              for j = 0:dbraw.nr_blocks.val(i)-1
49                  dbraw.file_name.val{j+1,i} = ...
50                  dlmread([dbraw.file_name.txt{i},'.txt'],'',...
51                  [126+block_length*j+26*j 0 126+block_length*(j+1)+26*j...
                      1]);
52                  count = count + 1
53              end
54          end
55      end
56      save(RawDataBaseFileName,'dbraw');
57      disp('dbraw created')
58  elseif  LoadRawDataBase
59      clear db_1
60      load(RawDataBaseFileName);
61      disp('dbraw loaded')
62  end
63  %...
      ========================================================================...

64  % Either moving data from raw database into a working database, adding...
       some
65  % general data and doing some general calculations or loading the ...
      working
66  % database from a previously made MAT file
67  if BuildWorkDataBase
68      clear db_1
69
70      % Genneral data
71      db_1.totaltest = length(dbraw.test_nr.val);
72      db_1.nr_samples = dbraw.nr_samples.val;
73      db_1.nr_blocks = dbraw.nr_blocks.val;
74      db_1.block_length = db_1.nr_samples./db_1.nr_blocks;
75      db_1.N_half = floor(db_1.block_length/2);
76      db_1.nu = 0.00001544;
77      disp('genneral data')
78
79      % Data from dbraw put into db_1
80      db_1.test_nr = dbraw.test_nr.val;
81      db_1.x_pos = dbraw.x_pos.val;
82      db_1.y_pos = dbraw.y_pos.val;
83      db_1.T_atm = dbraw.T_atm.val;
84      db_1.p_atm = dbraw.p_atm.val;
85      db_1.orifice_Δ_p = dbraw.orifice_Δ_p.val;
```

```
86      db_1.sampling_rate = dbraw.sampling_rate.val;
87
88      % Time and velocity data to db_1
89      for i = 1:db_1.totaltest
90          for j = 1:db_1.nr_blocks(i)
91              time_b(:,j,i) = dbraw.file_name.val{j,i}(:,1);
92              V_b(:,j,i) = dbraw.file_name.val{j,i}(:,2);
93          end
94          db_1.time(:,i) = time_b(:,1,i); % Takes only first time -> ...
              constant block time
95          db_1.V(:,i) = mean(V_b(:,:,i)');
96          db_1.dt(i) = db_1.time(2,i) - db_1.time(1,i);
97      end
98      disp('dbraw to db_1')
99
100     % Statistics
101     for i = 1:db_1.totaltest
102         for j = 1:db_1.nr_blocks(i)
103             V_avg_b(j,i) = sum(V_b(:,j,i))/db_1.block_length(i);
104             fluc_b(:,j,i) = V_b(:,j,i)-V_avg_b(j,i); ...
                                                % Flucturations
105             V_rms_b(j,i) = sqrt(sum(fluc_b(:,j,i).^2)/(...
                db_1.block_length(i)-1));    % Standart diviation
106             var_b(j,i) = V_rms_b(j,i)^2; ...
                                                        % Variance
107             RTI_b(j,i) = V_rms_b(j,i)./V_avg_b(j,i); ...
                                                % Relativ turbulence ...
                  intensity
108         end
109         db_1.V_avg(i) = mean(V_avg_b(:,i));
110         db_1.fluc(:,i) = mean(fluc_b(:,:,i)');
111         db_1.V_rms(i) = mean(V_rms_b(:,i));
112         db_1.var(i) = mean(var_b(:,i));
113         db_1.RTI(i) = mean(RTI_b(:,i));
114     end
115     disp('statistics')
116
117     % Intergrale length and time scale
118     for i = 1:db_1.totaltest
119         for j = 1:db_1.nr_blocks(i)
120             R_b(:,j,i) = autocorr(fluc_b(:,j,i),db_1.block_length(i)...
                -1);
121         end
122         db_1.R(:,i) = mean(R_b(:,:,i)');
123         db_1.R_x(:,i) = linspace(0,db_1.time(end,i),db_1.block_length(...
              i));
124
125         % Corregating for noise
126         stop = 7;
127         p = polyfit(db_1.R_x(2:stop),db_1.R(2:stop),2);
128         db_1.y = p(1)*db_1.R_x(1:end).^2+p(2)*db_1.R_x(1:end)+p(3);
129         db_1.R(1) = db_1.y(1);
130
131         stop_index = db_1.R(:,i) < 0;
132         stop = find(stop_index,1,'first');
133         db_1.T_scale(i) = trapz(db_1.R_x(1:stop,i),db_1.R(1:stop,i));
134         db_1.L_scale(i) = db_1.V_avg(i)*db_1.T_scale(i);
135     end
136     disp('intergral length and time scale')
```

```
137
138     % Kolmogorov length and time scale
139     for i = 1:db_1.totaltest
140         for j = 1:db_1.nr_blocks
141             for k = 2:db_1.block_length-1
142                 dudt(k,j,i) = (fluc_b(k+1,j,i)-fluc_b(k-1,j,i))/...
                      db_1.dt(i);
143             end
144         end
145         db_1.lambda(i) = db_1.V_avg(i)*db_1.V_rms/...
146                          sqrt(mean(mean(dudt(:,:,i).^2)));
147         db_1.epsilon(i) = 15*db_1.nu*db_1.V_rms^2/db_1.lambda(i)^2;
148         db_1.kol_L_scale(i) = (db_1.nu^3/db_1.epsilon(i))^(1/4);
149         db_1.kol_T_scale(i) = (db_1.nu/db_1.epsilon(i))^(1/2);
150         db_1.kol_v(i) = (db_1.epsilon(i)*db_1.nu)^(1/4);
151         db_1.kol_f(i) = db_1.V_avg(i)/(2*pi*db_1.kol_L_scale(i));
152     end
153     disp('kolmogorov length and time scale')
154
155     % Frequenzy analysis
156     for i = 1:db_1.totaltest
157         for j = 1:db_1.nr_blocks
158             V_fft_b(:,j,i) = db_1.dt(i)*fft(fluc_b(:,j,i));
159             fft_length_b(j,i) = length(V_fft_b(:,1,i));
160             V_mag_b(:,j,i) = abs(V_fft_b(:,j,i));
161             E_b(:,j,i) = V_mag_b(:,j,i).^2/db_1.time(end,i);
162         end
163         db_1.E(:,i) = mean(E_b(:,:,i)');
164         db_1.fft_length(i) = fft_length_b(1,i);
165
166         db_1.f(:,i) = linspace(0,db_1.sampling_rate(i),db_1.fft_length...
              (i));
167         db_1.A_E(i) = 2*trapz(db_1.f(1:db_1.N_half(i),i),...
168                       db_1.E(1:db_1.N_half(i),i));
169     end
170     disp('frequenzy analysis')
171
172     % Making an index for gartering data-sets with same y position
173     count = 0;
174     for i = 1:db_1.totaltest
175         if  i == 1
176             count = count + 1;
177             db_1.y_index(count) = db_1.y_pos(i);
178         elseif  sum(db_1.y_index(:) == db_1.y_pos(i)) == 0
179             count = count + 1;
180             db_1.y_index(count) = db_1.y_pos(i);
181         end
182     end
183     disp('y index')
184
185     %Save working database
186     save(WorkDataBaseFileName,'db_1','-v7.3');
187     disp('db_1 created')
188 elseif  LoadWorkDataBase
189     load(WorkDataBaseFileName);
190     disp('db_1 loaded')
191 end
```

Appendix D.2

Function for strippping Excel sheet

```
function dbraw = excelstrip(ExcelProperties,VariableList1,...
 VariableList2,xlsversion)
%
% Strips data from a list of excel workbooks and collect all the data ...
 into a
% single data structure (dbraw). The excel sheet has to follow some ...
 standards:
%   - It should contain two tables:
%       1)  The first table should have one header row and below that,...
  one data row.
%           The width of the table is unlimited and it does not have ...
 to start
%           in first column of the sheet.
%       2)  The second table should have one header row and below that...
 , unlimited
%           number of data rows. The width of the table is unlimited ...
 but it
%           should start in the same column as the first table.
%   - The header cells should not contain nummeric data. The data ...
 cells may contain
%     nummeric data or text. Empty cells within the table will become ...
 NaN and [] in
%     the data structure (dbraw);
%
% ...
 ================================================================================...

% Arguments
%
% Excelproperties: Cell array containing the filename of the excel ...
 workbook (with
% or without path) in the first column. Sheetname or index number of ...
 worksheet in
% the second row. Third and fourth column should contain the index ...
 number for the
% first and second table header row, respectively.
%
% Example:
% ExcelProperties = {
% 'Group 1\HCCI test scheme1200 rpm.xls'                    'Sheet1'...
          2       5
% 'Group 1\HCCI test scheme1200 rpm.xls'                    'Sheet2'...
          2       5
% 'Group 3\HCCI test scheme1800 rpm.xls'                    'Sheet1'...
          2       5
% 'Group 3\HCCI test scheme1800 rpm.xls'                    'Sheet2'...
          2       5
% '..\last year\Group 3\HCCI test scheme1800 rpm.xls'       1       ...
          4       9
% '..\last year\Group 3\HCCI test scheme1800 rpm.xls'       2       ...
          4       9
% '..\last year\Group 3\HCCI test scheme1800 rpm.xls'       3       ...
          4       9
```

```
33 % };
34 %
35 % VariableList1: Should contain a cell array with variable names that ...
     you wish to give
36 % the column data from the first table of the sheet. VariableList1 ...
     should have at least
37 % same length as the first table in the sheet with the widest table.
38 %
39 % Example:
40 % VariableList1 = {
41 % 'operator','T_atm','p_atm','date','N','CR','fuel','pressure_file_dir...
     ','motoring_file'
42 % };
43 %
44 % VariableList2: Same as VariableList1 but with regard to the second ...
     table.
45 %
46 % xlsversion: Options are 'xls97' and 'xls95'. 'xls95' is ¬5x faster ...
     than 'xls97'
47 % but also less flexible regarding data compatibility.
48 %
49 % MAC users has to run xls95 and make sure that the workbook is saved ...
     as so. To
50 % ensure this save the sheet as 95 version xls, close the sheet and ...
     reopen
51 % and save it again...... Don't ask me why it is necessary to save it ...
     two times
52 % as 95 version to enforce it to happen. Ask Bill Gates.
53 %
54 % ...
     ======================================================================...

55 % Output
56 % dbraw is a single data structure containing all the data from all ...
     the
57 % the specified excelsheets.
58 %
59 % dbraw will contain substructures with the names specified in ...
     VariableList1
60 % and VariableList2. Each substructure contain three fields: head, txt...
      and val.
61 %   -head is a cell array of header text from the respective sheet and...
      column.
62 %   -txt is a cell array of cell text from the respective sheet, ...
     column and row.
63 %   -val is a vector with nummeric value from the respective sheet, ...
     column and row.
64 %
65 % In addition to the substructures specified in VariableList1 and ...
     VariableList2
66 % fields are made with the names: GeneralDataHeader, GeneralDataTxt, ...
     GeneralDataVal,
67 % TestHeader, TestDataTxt, TestDataVal.
68 % They contain large cell arrays with all sheet content as either ...
     header text,
69 % cell text or nummeric values. GeneralData means data from the first ...
     table in
70 % the excel sheets. TestData means data from the second table. General...
      data from
```

```
71 % each sheet has been copied to fit the number of test data rows.
72 %
73 % A field called xlsFileName is made. It contains the filename with ...
     partial path
74 % relative to the working directory.
75 % excelsheet from which the row data has been stripped.
76 %
77 % A field called WorkingDir contain the full path of the working ...
     directory
78 % where the matlab script was run.
79
80 TestDataVal = {};
81 TestDataTxt = {};
82 TestDataHeader = {};
83 TestDataSizes = [];
84
85 GeneralDataVal = {};
86 GeneralDataTxt = {};
87 GeneralDataHeader = {};
88 GeneralDataSizes = [];
89
90 xlsFileName = {};
91 xlsSheet = {};
92
93 for i = 1:size(ExcelProperties,1)
94
95     sheet = ExcelProperties{i,2};
96     header1row = ExcelProperties{i,3};
97     header2row = ExcelProperties{i,4};
98
99     if strcmp(xlsversion,'xls95')
100        [dum,dum,raw] = xlsread(ExcelProperties{i,1},sheet,'A1:B2','...
             basic');
101    elseif strcmp(xlsversion,'xls97')
102        [dum,dum,raw] = xlsread(ExcelProperties{i,1},sheet);
103    else
104        [dum,dum,raw] = xlsread(ExcelProperties{i,1},sheet);
105    end
106
107    raw1 = raw(header1row:header1row+1,:);
108    textcells1 = cellfun(@isstr,raw1);
109    raw1strless = raw1;
110    raw1strless(textcells1) = {'-'};
111    emptycells1 = cellfun(@isnan,raw1strless);
112    valcells1 = ¬emptycells1 & ¬textcells1;
113    firstcol = find(any(¬emptycells1,1),1,'first');
114    lastcol = find(any(¬emptycells1,1),1,'last');
115    raw1 = raw1(:,firstcol:lastcol);
116    emptycells1 = emptycells1(:,firstcol:lastcol);
117    textcells1 = textcells1(:,firstcol:lastcol);
118    valcells1 = ¬emptycells1 & ¬textcells1;
119
120
121    raw2 = raw(header2row:end,:);
122    textcells2 = cellfun(@isstr,raw2);
123    raw2strless = raw2;
124    raw2strless(textcells2) = {'-'};
125    emptycells2 = cellfun(@isnan,raw2strless);
126    firstcol = find(any(¬emptycells2,1),1,'first');
```

```
127     lastcol = find(any(¬emptycells2,1),1,'last');
128     lastrow = find(any(¬emptycells2,2),1,'last');
129     raw2 = raw2(1:lastrow,firstcol:lastcol);
130     emptycells2 = emptycells2(1:lastrow,firstcol:lastcol);
131     textcells2 = textcells2(1:lastrow,firstcol:lastcol);
132     valcells2 = ¬emptycells2 & ¬textcells2;
133
134     headerraw = raw2(1,:);
135     dataraw = raw2(2:end,:);
136     TestDataSizes = [TestDataSizes; size(dataraw)];
137
138     TestDataHeader{i} = cell(headerraw);
139     TestDataHeader{i}(textcells2(1,:)) = headerraw(textcells2(1,:));
140     TestDataHeader{i} = repmat(TestDataHeader{i},TestDataSizes(i,1),1)...
          ;
141
142     TestDataVal{i} = nan(size(dataraw));
143     TestDataVal{i}(valcells2(2:end,:)) = [dataraw{valcells2(2:end,:)...
          }];
144
145     TestDataTxt{i} = cell(size(dataraw));
146     TestDataTxt{i}(textcells2(2:end,:)) = dataraw(textcells2(2:end,:))...
          ;
147
148     headerraw = raw1(1,:);
149     dataraw = raw1(2,:);
150     GeneralDataSizes = [GeneralDataSizes; size(dataraw)];
151
152     GeneralDataHeader{i} = cell(headerraw);
153     GeneralDataHeader{i}(textcells1(1,:)) = headerraw(textcells1(1,:))...
          ;
154     GeneralDataHeader{i} = repmat(GeneralDataHeader{i},TestDataSizes(i...
          ,1),1);
155
156     GeneralDataVal{i} = nan(size(dataraw));
157     GeneralDataVal{i}(valcells1(2,:)) = [dataraw{valcells1(2,:)}];
158     GeneralDataVal{i} = repmat(GeneralDataVal{i},TestDataSizes(i,1),1)...
          ;
159
160     GeneralDataTxt{i} = cell(size(dataraw));
161     GeneralDataTxt{i}(textcells1(2,:)) = dataraw(textcells1(2,:));
162     GeneralDataTxt{i} = repmat(GeneralDataTxt{i},TestDataSizes(i,1),1)...
          ;
163
164     xlsFileName{i} = repmat(ExcelProperties(i,1),TestDataSizes(i,1),1)...
          ;
165     xlsSheet{i} = repmat(ExcelProperties(i,2),TestDataSizes(i,1),1);
166
167 end
168
169 dbraw.general.head = {};
170 dbraw.general.txt = {};
171 dbraw.general.val = [];
172
173 dbraw.test.head = {};
174 dbraw.test.txt = {};
175 dbraw.test.val = [];
176
177 dbraw.xlsFileName = {};
```

```
178 dbraw.xlsSheet = {};
179
180 GeneralMaxRows = max(GeneralDataSizes(:,2));
181 TestMaxRows = max(TestDataSizes(:,2));
182
183 for i = 1:size(ExcelProperties,1)
184
185     dbraw.general.head = [dbraw.general.head;...
186         [GeneralDataHeader{i} cell(TestDataSizes(i,1),...
187          GeneralMaxRows-GeneralDataSizes(i,2))]];
188     dbraw.general.txt = [dbraw.general.txt; ...
189         [GeneralDataTxt{i} cell(TestDataSizes(i,1),...
190          GeneralMaxRows-GeneralDataSizes(i,2))]];
191     dbraw.general.val = [dbraw.general.val; ...
192         [GeneralDataVal{i} nan(TestDataSizes(i,1),...
193          GeneralMaxRows-GeneralDataSizes(i,2))]];
194
195     dbraw.test.head = [dbraw.test.head;...
196         [TestDataHeader{i} cell(TestDataSizes(i,1),...
197          TestMaxRows-TestDataSizes(i,2))]];
198     dbraw.test.txt = [dbraw.test.txt; ...
199         [TestDataTxt{i} cell(TestDataSizes(i,1),...
200          TestMaxRows-TestDataSizes(i,2))]];
201     dbraw.test.val = [dbraw.test.val; ...
202         [TestDataVal{i} nan(TestDataSizes(i,1),...
203          TestMaxRows-TestDataSizes(i,2))]];
204
205     dbraw.xlsFileName = [dbraw.xlsFileName; xlsFileName{i}];
206     dbraw.xlsSheet = [dbraw.xlsSheet; xlsSheet{i}];
207
208 end
209
210 for i = 1:length(VariableList1)
211
212     dbraw = setfield(dbraw,VariableList1{i},..
213             struct('head',{dbraw.general.head(:,i)},...
214             'txt',{dbraw.general.txt(:,i)},'val',{dbraw.general.val(:,...
                  i)}));
215
216 end
217
218 for i = 1:length(VariableList2)
219
220     dbraw = setfield(dbraw,VariableList2{i},...
221             struct('head',{dbraw.test.head(:,i)},...
222             'txt',{dbraw.test.txt(:,i)},'val',{dbraw.test.val(:,i)}));
223
224 end
225
226     wd = what;
227     dbraw.WorkingDir = wd.path;
```

Appendix D.3

Script for collecting data bases

Script for collection of data bases is here presented.

```
clear all
close all
clc

standard_plots = 0;
scales_plots = 0;
autocorrelation = 1;
energy_plots  = 0;

% ===== nomeclatur =====
%   1 ¬ 10_0500
%   2 ¬ 10_0675
%   3 ¬ 10_0850
%   4 ¬ m5_0500
%   5 ¬ m5_0675
%   6 ¬ m5_0850

% Loading data-bases
load('db_1');
load('db_2');
load('db_3');
load('db_4');
load('db_5');
load('db_6');

my_colormap = [0 0.4470 0.7410; 0.8500 0.3250 0.0980; 0.9290 0.6940 0...
  .1250;...
             0.4940 0.1840 0.5560; 0.4660 0.6740 0.1880; 0.3010 0.7450...
                0.9330];

if standard_plots
    V_avg_plots(db_1,db_2,db_3,db_4,db_5,db_6,my_colormap)
    RTI_plots(db_1,db_2,db_3,db_4,db_5,db_6,my_colormap)
end
if scales_plots
    taylor_microscale(db_1,db_2,db_3,db_4,db_5,db_6,my_colormap)
    intergral_time_and_length_scales_plots(db_1,db_2,db_3,db_4,db_5,...
      db_6,...
        my_colormap)
    kolmogorov_time_and_length_scales_plots(db_1,db_2,db_3,db_4,db_5,...
      db_6,...
        my_colormap)
end
if autocorrelation
   autocorrelation_plots(db_1,db_2,db_3,db_4,db_5,db_6,my_colormap)
end
if energy_plots
    energy_spectrum_plots(db_1,db_2,db_3,db_4,db_5,db_6,my_colormap)
end
```

Appendix E

Energy spectrum with all scales

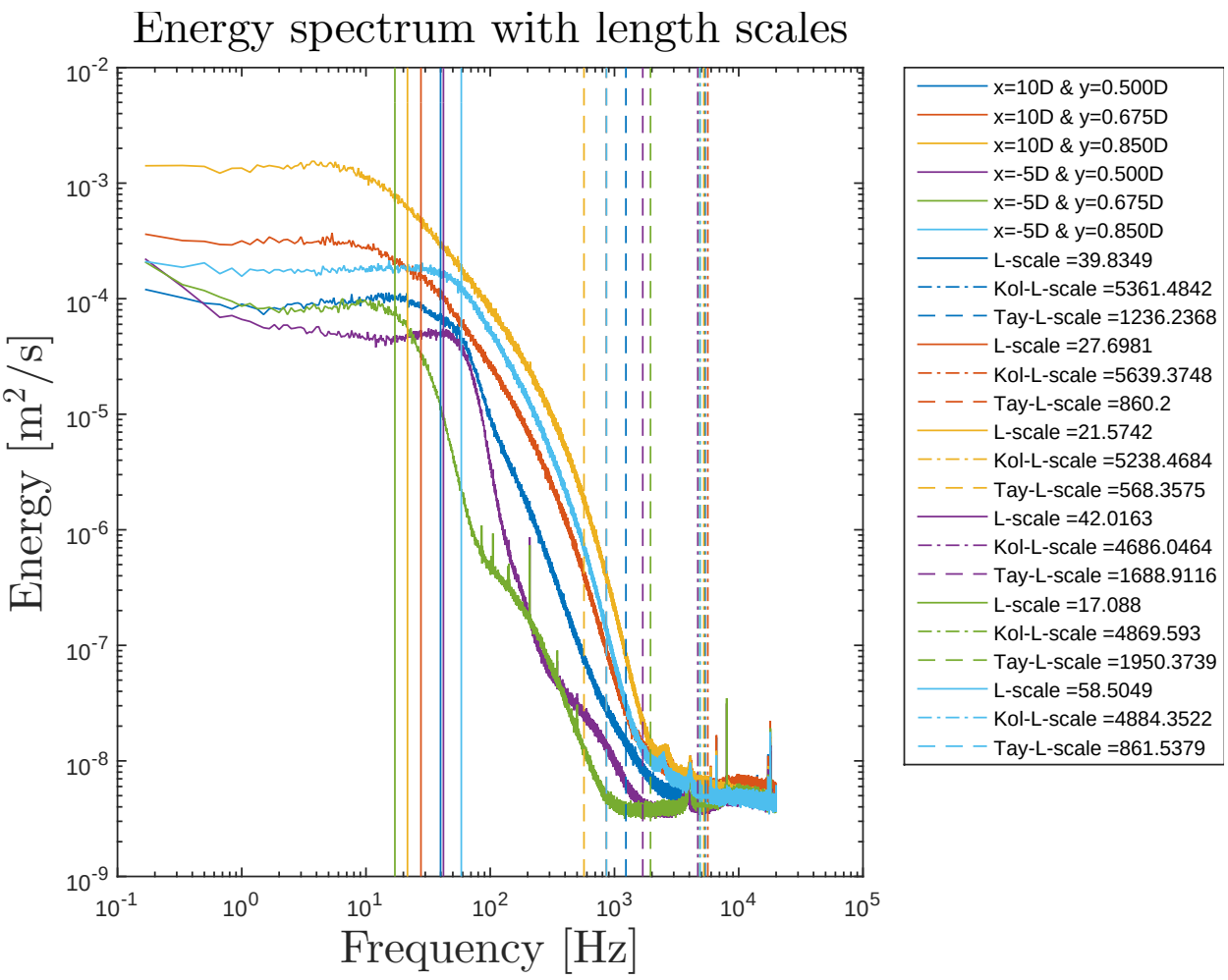
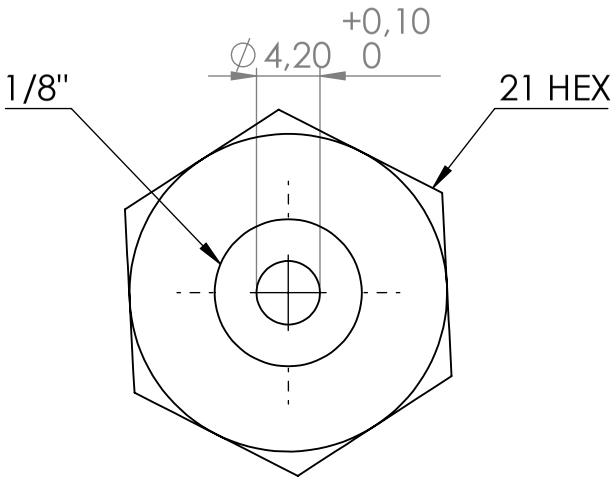
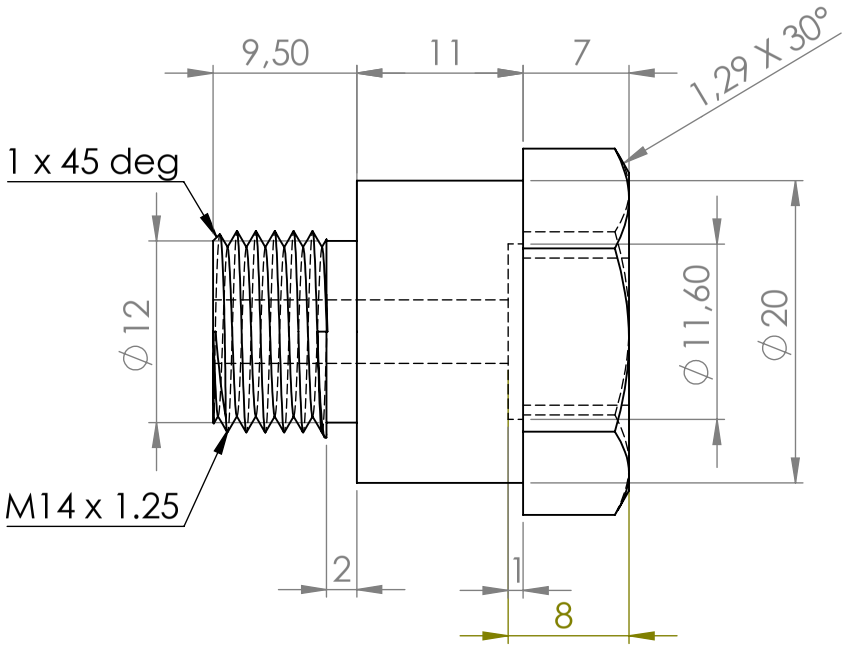


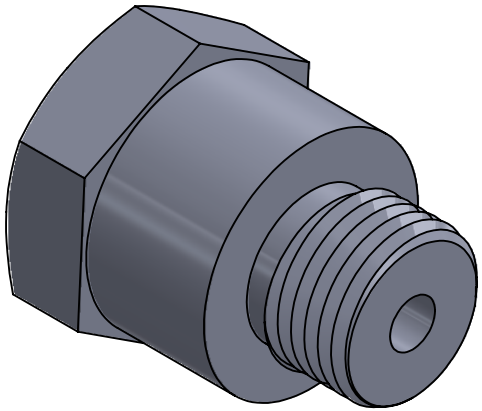
Figure 69: Energy spectrum with all scales

Appendix F

Working drawing of adapter







Material:	Stainless steel
Contacts:	Rasmus Katkjær - s103380 tlf: 30244521
TITLE:	Adapter
DWG NO.	adapter_drawing_10
A4	
SCALE:2:1	SHEET 1 OF 1

Appendix G

MatLab script for data acquisition

```
clear all
close all
clc

% ========================================================================%
% User input:
% ========================================================================%

foldername = 'DAQ_data_full';
filename = 'final';
data_name = '0008_90_1000_0';

samplerate = 100000;     %[Hz]
cycles = 500;           %[2*omd]
N = 1000;               %[rpm]

% Session data
sampleduration = cycles*2/(N/60)         %[s]
number_samples = samplerate*sampleduration;  %[-]

% ========================================================================%
% Setting up the data acquisition engine:
% ========================================================================%

session = daq.createSession('ni');
chan = addAnalogInputChannel(session,'Dev1',[0:1],'Voltage');
session.Rate = samplerate;
samplerate = session.Rate;
session.DurationInSeconds = sampleduration;

sampleduration = session.DurationInSeconds;
samplespertrigger = sampleduration/samplerate;

chan(1).Range = [-10 10];

chan(2).Range = [-10 10];

% ========================================================================%
% Running acquisition and saving data:
% ========================================================================%

disp('Data acqusition started')
[DAQdata_measured,timestamp_measured] = startForeground(session);

hw_v = DAQdata_measured(:,1);
rev_v = DAQdata_measured(:,2);
timestamp = timestamp_measured(:);


    %timestamp(1+number_samples*(i-1):number_samples+number_samples*(i...
      -1)) = timestamp_measured(:);

disp('Data acqusition ended')
```

```
53
54 save([foldername,'\',filename,'_',data_name,'.mat'],'hw_v','rev_v','...
     timestamp','-mat')
55
56 % ======================================================================%
57 % Plotting:
58 % ======================================================================%
59
60 plotting = 0;
61
62 if plotting
63     figure(1)
64     plot(timestamp,hw_v,timestamp,rev_v)
65     ylim([0 10])
66
67     %figure(2)
68     %plot(timestamp,DAQdata(:,1),timestamp,DAQdata(:,2))
69
70 end
```

Appendix H

Excel file with data and file names for the engine test

Genneral comments
Date: 21-02-2016

# Test run	Depth [m]	Rotation [deg]	Engine speed [rpm]	Throttel [%]	Temp. (atm) [C]	Press. (atm) [Pa]	Sampling rate [Hz]	Sample duration [s]	Bridge res. [ohm]	Calibration file	Filename	Comment
1	0.004	0	600	1	18.5	99232	100000	100	738	cal_final	final_0004_0_600_1	
2	0.004	0	1000	1	18.4	99232	100000	60	738	cal_final	final_0004_0_1000_1	
3	0.004	0	1500	1	18.5	99232	100000	40	738	cal_final	final_0004_0_1500_1	
4	0.004	0	500	1	18.4	99232	100000	120	738	cal_final	final_0004_0_500_1	
5	0.004	0	1000	100	18.5	99232	100000	60	738	cal_final	final_0004_0_1000_100	
6	0.004	0	1000	75	18.5	99232	100000	60	738	cal_final	final_0004_0_1000_75	
7	0.004	0	1000	50	18.5	99232	100000	60	738	cal_final	final_0004_0_1000_50	
8	0.004	0	1000	25	18.5	99232	100000	60	738	cal_final	final_0004_0_1000_25	
9	0.004	0	1000	0	18.5	99232	100000	60	738	cal_final	final_0004_0_1000_0	
10	0.004	90	500	1	18.3	99232	100000	120	738	cal_final	final_0004_90_500_1	hall sensor ustabil
14	0.004	90	1000	1	18.4	99232	100000	60	738	cal_final	final_0004_90_1000_1	
12	0.004	90	1500	1	18.5	99232	100000	40	738	cal_final	final_0004_90_1500_1	
14	0.004	90	1000	100	18.5	99232	100000	60	738	cal_final	final_0004_90_1000_100	
15	0.004	90	1000	75	18.5	99232	100000	60	738	cal_final	final_0004_90_1000_75	
16	0.004	90	1000	50	18.4	99232	100000	60	738	cal_final	final_0004_90_1000_50	
17	0.004	90	1000	25	18.3	99232	100000	60	738	cal_final	final_0004_90_1000_25	
18	0.004	90	1000	0	18.3	99232	100000	60	738	cal_final	final_0004_90_1000_0	
19	0.008	0	500	1	18.3	99232	100000	120	738	cal_final	final_0008_0_500_1	Cabele twisted before run. Voltage up to 7v. Mented
20	0.008	0	1000	1	18.4	99232	100000	60	738	cal_final	final_0008_0_1000_1	
21	0.008	0	1500	1	18.3	99232	100000	40	738	cal_final	final_0008_0_1500_1	
22	0.008	0	1000	100	18.3	99232	100000	60	738	cal_final	final_0008_0_1000_100	
23	0.008	0	1000	75	18.3	99232	100000	60	738	cal_final	final_0008_0_1000_75	
24	0.008	0	1000	50	18.2	99232	100000	60	738	cal_final	final_0008_0_1000_50	
25	0.008	0	1000	25	18.2	99232	100000	60	738	cal_final	final_0008_0_1000_25	
26	0.008	0	1000	0	18.2	99232	100000	60	738	cal_final	final_0008_0_1000_0	
27	0.008	90	500	1	18.3	99232	100000	60	738	cal_final	final_0008_90_500_1	
28	0.008	90	1000	1	18.3	99232	100000	40	738	cal_final	final_0008_90_1000_1	
29	0.008	90	1500	1	18.2	99232	100000	120	738	cal_final	final_0008_90_1500_1	
30	0.008	90	1000	100	18.2	99232	100000	60	738	cal_final	final_0008_90_1000_100	
31	0.008	90	1000	75	18.2	99232	100000	60	738	cal_final	final_0008_90_1000_75	
32	0.008	90	1000	50	18.2	99232	100000	60	738	cal_final	final_0008_90_1000_50	
33	0.008	90	1000	25	18.2	99232	100000	60	738	cal_final	final_0008_90_1000_25	
34	0.008	90	1000	0	18.2	99232	100000	60	738	cal_final	final_0008_90_1000_0	

Appendix I

MatLab script for engine test

The main script for creating the data bases can be found here. Again, only one of the 3 scripts are shown. The only thing which differ between the scripts are the name of the data bases, the Excel sheet and the check for choosing the correct engine revolution. The same script for stripping the excel was used as for the pipe test.

Appendix I.1

Main engine script

```
clear all
close all
clc

BuildRawDataBase = 0;
LoadRawDataBase = 1;
RawDataBaseFileName = 'dbraw';

BuildWorkDataBase = 1;
LoadWorkDataBase = 0;
WorkDataBaseFileName = 'db_2';

xlsversion = 'xls97'; % Options are 'xls97' and 'xls95'. MAC shoud use...
   'xls95'

%...
  ========================================================================...

% Either loading all data from excelsheets and LabView files into a
% Matlab structure or from a preveously generated Mat file
if BuildRawDataBase
    clear db_2

    % Properties of the excelsheets that should be stripped
    ExcelProperties = {
    % (path\)filename                       1st header row     2nd ...
      header row
    'final_scheme_2_test'  'Ark1'           3                  5
    };

    % list of variable names for the general data belonging to the 1st...
       header
    VariableList1 = {
    'date'
    };

    % list of variable names for the test data belonging to the 2nd ...
      header
    VariableList2 = {
    'test_nr','dep','rot','rpm','throttle','T_atm','p_atm','SR','SD','...
      R_bridge',...
    'cal_file','file_name','comment'
```

```
36      };
37
38      % stripping all excel data into a single structure
39      dbraw = excelstrip(ExcelProperties,VariableList1,VariableList2,...
          xlsversion);
40
41      % adding time-voltage files to the structure
42      dbraw.file_name.val = cell(size(dbraw.file_name.txt));
43
44      % adding calibrations files to the structure
45      dbraw.cal_file.val = cell(size(dbraw.cal_file.txt));
46
47      count = 0;
48      for i = 1:length(dbraw.file_name.txt)
49          if ¬isempty(dbraw.file_name.txt{i})
50              dbraw.file_name.val{i} =...
51                  load(['DAQ_data','\',dbraw.file_name.txt{i},'.mat']);
52          end
53      end
54
55      count = 0;
56      for i = 1:length(dbraw.cal_file.txt)
57          if ¬isempty(dbraw.cal_file.txt{i})
58              dbraw.cal_file.val{i} =...
59                  dlmread(['DAQ_data','\',dbraw.cal_file.txt{i},'.txt'])...
                      ;
60          end
61      end
62
63      save(RawDataBaseFileName,'dbraw');
64      disp('dbraw created')
65  elseif  LoadRawDataBase
66      clear db_2
67      load(RawDataBaseFileName);
68      disp('dbraw loaded')
69  end
70
71  %...
      ========================================================================...

72  % Either moving data from raw database into a working database, adding...
       some
73  % general data and doing some general calculations or loading the ...
      working
74  % database from a previously made MAT file
75  if BuildWorkDataBase
76      clear db_2
77
78      % === CAD SWITCH CHECK ======
79      CAD_switch = [1 1 1 1];
80      test_of_switch = 0;
81
82      % Genneral data
83      db_2.totaltest = length(dbraw.test_nr.val);
84      db_2.nu = 0.00001529;
85      db_2.nu_mean = 9.8170e-06;
86      db_2.int_nr = 8;
87      db_2.CAD_adj = 135;                     %[CAD]
88      db_2.CAD_int = 360/db_2.int_nr;           %[CAD]
```

```
89      disp('genneral data')
90
91      % Data from dbraw put into db
92      db_2.test_nr = dbraw.test_nr.val;
93      db_2.dep = dbraw.dep.val;
94      db_2.rot = dbraw.rot.val;
95      db_2.rpm = dbraw.rpm.val;
96      db_2.T_atm = dbraw.T_atm.val+273.15;      %[K]
97      db_2.p_atm = dbraw.p_atm.val;             %[Pa]
98      db_2.R_bridge = dbraw.R_bridge.val;                 %[ohm]
99      db_2.SR = dbraw.SR.val;
100     db_2.SD = dbraw.SD.val;
101
102     % Time and velocity data to db
103     for i = 1:db_2.totaltest
104         db_2.time(:,i) = dbraw.file_name.val{i}.timestamp;
105         db_2.hw_v(:,i) = dbraw.file_name.val{i}.hw_v;
106         db_2.rev_v(:,i) = dbraw.file_name.val{i}.rev_v;
107         db_2.dt(i) = db_2.time(2,i) - db_2.time(1,i);
108     end
109
110     % Calibrations points to db
111     for i = 1:db_2.totaltest
112         db_2.cal_V(:,i) = dbraw.cal_file.val{i}(:,1);
113         db_2.cal_E(:,i) = dbraw.cal_file.val{i}(:,2);
114     end
115
116     % Test data
117     for i = 1:db_2.totaltest
118         cyc(i) = db_2.SD(i)/60*db_2.rpm(i);
119     end
120
121     max_cyc_length = ceil(max(db_2.SD)/min(cyc)*max(db_2.SR)*1.1);
122     disp('dbraw to db')
123
124
125     % Cutting of voltage signal into closed revolutions
126     hw_v_cut = zeros(max_cyc_length,400,db_2.totaltest);
127     for i = 1:db_2.totaltest
128        [hw_v_cut_now,cut_length(i),db_2.cut_nr(i)] = cyc_cut(db_2.hw_v...
            (:,i),db_2.rev_v(:,i),db_2.CAD_adj,db_2.rpm(i),db_2.SR(i),...
            CAD_switch(i));
129        hw_v_cut(1:cut_length(i),1:db_2.cut_nr(i),i) = hw_v_cut_now;
130     end
131     disp('Cut into cycles')
132
133     % Simulating engine pressure and temperature
134     p = zeros(max_cyc_length,max(cyc),db_2.totaltest);
135     T = zeros(max_cyc_length,max(cyc),db_2.totaltest);
136     CAD = zeros(max_cyc_length,max(cyc),db_2.totaltest);
137     for i = 1:db_2.totaltest
138         for j = 1:db_2.cut_nr(i)
139             index = hw_v_cut(:,j,i) ~= 0;
140             [p(index,j,i),T(index,j,i),CAD(index,j,i)] = p_v_sim(...
                  db_2.rpm(i),sum(index),db_2.p_atm(i),db_2.T_atm(i));
141         end
142     end
143     disp('Pressure and Tempreture simulated')
144
```

```
145     % Temperature correction
146     db_2.V = zeros(max_cyc_length,max(cyc),db_2.totaltest);
147     for i = 1:db_2.totaltest
148         for j = 1:db_2.cut_nr
149             index = hw_v_cut(:,j,i) ~= 0;
150             [db_2.V(index,j,i)] = T_corr(hw_v_cut(index,j,i),p(index,j...
                  ,i),T(index,j,i),db_2.cal_V(:,i),db_2.cal_E(:,i),db_2.nu,...
                  db_2.R_bridge(i),db_2.T_atm(i));
151         end
152     end
153     disp('Velocity found from temp. correcton')
154
155     if test_of_switch
156         figure(1)
157         for i = 1:db_2.totaltest
158             index = db_2.V(:,1,i) ~= 0;
159             subplot(db_2.totaltest,1,i)
160             plot(CAD(index,1,i),db_2.V(index,1,i))
161             ylim([0 50])
162         end
163         return
164     end
165
166     % Cutting of velosity signal into CAD intervals
167     for i = 1:db_2.totaltest
168         for j = 1:db_2.cut_nr
169             index = hw_v_cut(:,j,i) ~= 0;
170             [V_int,int_CAD,int_length(i)] = int_cut(db_2.V(index,j,i),...
                  db_2.CAD_int,CAD(:,j,i),db_2.int_nr);
171             db_2.V_int(1:int_length(i),1:db_2.int_nr,j,i) = V_int(1:...
                  int_length,1:db_2.int_nr);
172             db_2.int_CAD(1:int_length(i),1:db_2.int_nr,j,i) = int_CAD...
                  (1:int_length,1:db_2.int_nr);
173         end
174     end
175     disp('Cut into CAD intervals')
176
177 %     figure(110)
178 %     for i = 1:db_2.cut_nr
179 %         index = hw_v_cut(:,i,1) ~= 0;
180 %         subplot(db_2.cut_nr,1,i)
181 %         plot(CAD(index,i,1),db_2.V(index,i,1))
182 %     end
183 %
184 %     figure(111)
185 %      for i = 1:db_2.int_nr
186 %          k = 1;
187 %         index = db_2.V_int(:,i,k,1) ~= 0;
188 %         subplot(1,db_2.int_nr,i)
189 %         plot(db_2.int_CAD(1:sum(index),i,k,1),db_2.V_int(1:sum(index...
    ),i,k,1))
190 %         ylim([0 30])
191 %      end
192
193
194     % Statistics
195     bw = 2;
196     plotflag = 'plot_off';
197
```

```
198     for i = 1:db_2.totaltest
199         bandpasses = [200 db_2.SR(i) 1 1];
200         for j = 1:db_2.cut_nr
201             for k = 1:db_2.int_nr
202                 index = db_2.V_int(:,k,j,i) ≠ 0;
203                 V_avg_b(k,j,i) = sum(db_2.V_int(index,k,j,i))/sum(...
                      index);
204                                 t = linspace(0,1/db_2.SR(i)*length(...
                                      db_2.V_int(index,k,j,i)),length(...
                                      db_2.V_int(index,k,j,i)))';
205                 if floor(sum(index)/2) == sum(index)/2
206                     fluc_b(index,k,j,i) =fftBPfilter(t,db_2.V_int(...
                          index,k,j,i),bandpasses,bw,plotflag); ...
                                          % Flucturations
207                     V_rms_b(k,j,i) = sqrt(sum(fluc_b(index,k,j,i).^2)...
                          /(sum(index)-1));    % Standart diviation
208                 else
209                     fluc_b(1:sum(index)-1,k,j,i) =fftBPfilter(t(1:end...
                          -1),db_2.V_int(1:sum(index)-1,k,j,i),bandpasses,...
                          bw,plotflag);                  % Flucturations
210                     V_rms_b(k,j,i) = sqrt(sum(fluc_b(1:sum(index)-1,k,...
                          j,i).^2)/(sum(index)-1-1));    % Standart ...
                          diviation
211                 end
212                 var_b(k,j,i) = V_rms_b(k,j,i)^2; ...
                                                        % Variance
213                 RTI_b(k,j,i) = V_rms_b(k,j,i)./V_avg_b(k,j,i); ...
                                        % Relativ turbulence intensity
214             end
215         end
216     end
217
218     for i = 1:db_2.totaltest
219         for k = 1:db_2.int_nr
220 %             for l = 1:length(fluc_b)
221 %                 db_2.fluc(l,k,i) = mean(fluc_b(l,k,:,i),3); % don't ...
      need it and is put together wrong....
222 %             end
223             db_2.V_avg(k,i) = mean(V_avg_b(k,:,i));
224             db_2.V_rms(k,i) = mean(V_rms_b(k,:,i));
225             db_2.var(k,i) = mean(var_b(k,:,i));
226             db_2.RTI(k,i) = mean(RTI_b(k,:,i));
227         end
228     end
229     disp('statistics')
230
231     % Intergrale length and time scale
232     R_b = zeros(length(fluc_b),db_2.int_nr,max(db_2.cut_nr),...
          db_2.totaltest);
233     for i = 1:db_2.totaltest
234         for j = 1:db_2.cut_nr
235             for k = 1:db_2.int_nr
236                 index = fluc_b(:,k,j,i) ≠ 0;
237                 R_b(1:sum(index),k,j,i) = autocorr(fluc_b(index,k,j,i)...
                      ,sum(index)-1);
238             end
239         end
240     end
241
```

```
242     for i = 1:db_2.totaltest
243         for k = 1:db_2.int_nr
244             for l = 1:length(fluc_b)
245                 db_2.R(l,k,i) = mean(R_b(l,k,:,i),3);
246             end
247             db_2.R_x(:,k,i) = linspace(0,length(db_2.R(:,k,i))/db_2.SR...
                  (i),length(db_2.R(:,k,i)));
248
249             % Corregating for noise
250             stop = 7;
251             p = polyfit(db_2.R_x(2:stop,k,i),db_2.R(2:stop,k,i),2);
252             db_2.y(:,k,i) = p(1)*db_2.R_x(1:end,k,i).^2+p(2)*db_2.R_x...
                  (1:end,k,i)+p(3);
253 %           db_2.R(1,k,i) = db_2.y(1,k,i);
254
255             stop_index = db_2.R(:,k,i) < 0;
256             stop = find(stop_index,1,'first');
257             db_2.T_scale(k,i) = trapz(db_2.R_x(1:stop,k,i),db_2.R(1:...
                  stop,k,i));
258             db_2.L_scale(k,i) = db_2.V_avg(k,i)*db_2.T_scale(k,i);
259         end
260     end
261     disp('intergral length and time scale')
262
263     % Kolmogorov length and time scale
264     for i = 1:db_2.totaltest
265         for j = 1:db_2.cut_nr
266             for k = 1:db_2.int_nr
267                 for l = 2:length(fluc_b)-1
268                     dudt(l,k,j,i) = (fluc_b(l+1,k,j,i)-fluc_b(l-1,k,j,...
                          i))/db_2.dt(i);
269                 end
270             end
271         end
272     end
273
274     for i = 1:db_2.totaltest
275         for k = 1:db_2.int_nr
276             db_2.lambda(k,i) = db_2.V_avg(k,i)*db_2.V_rms(k,i)/...
277                                sqrt(mean(mean(mean(dudt(:,:,:,i).^2))));
278             db_2.epsilon(k,i) = 15*db_2.nu_mean*db_2.V_rms(k,i)^2/...
                  db_2.lambda(k,i)^2;
279             db_2.kol_L_scale(k,i) = (db_2.nu_mean^3/db_2.epsilon(k,i))...
                  ^(1/4);
280             db_2.kol_T_scale(k,i) = (db_2.nu_mean/db_2.epsilon(k,i))...
                  ^(1/2);
281             db_2.kol_v(k,i) = (db_2.epsilon(k,i)*db_2.nu_mean)^(1/4);
282             db_2.kol_f(k,i) = db_2.V_avg(k,i)/(2*pi*db_2.kol_L_scale(k...
                  ,i));
283         end
284     end
285
286     disp('kolmogorov length and time scale')
287
288     % Frequenzy analysis
289     for i = 1:db_2.totaltest
290         for j = 1:db_2.cut_nr
291             for k = 1:db_2.int_nr
292                 V_fft_b(:,k,j,i) = db_2.dt(i)*fft(fluc_b(:,k,j,i));
```

```
293                 fft_length_b(k,i) = length(V_fft_b(:,k,j,i));
294                 V_mag_b(:,k,j,i) = abs(V_fft_b(:,k,j,i));
295                 E_b(:,k,j,i) = V_mag_b(:,k,j,i).^2/(int_length(i)/...
                      db_2.SR(i));
296             end
297         end
298     end
299
300     for i = 1:db_2.totaltest
301         for k = 1:db_2.int_nr
302             db_2.E(:,k,i) = mean(E_b(:,k,:,i),3);
303             db_2.fft_length(k,i) = fft_length_b(k,i);
304
305             db_2.f(:,k,i) = linspace(0,db_2.SR(i),db_2.fft_length(k,i)...
                  );
306             db_2.A_E(k,i) = 2*trapz(db_2.f(1:length(floor(db_2.f(:,k,i...
                  )/2)),k,i),db_2.E(1:length(floor(db_2.f(:,k,i)/2)),k,i));
307
308         end
309     end
310     disp('frequenzy analysis')
311
312     %Save working database
313     save(WorkDataBaseFileName,'db_2','-v7.3');
314     disp('db_2 created')
315 elseif  LoadWorkDataBase
316     load(WorkDataBaseFileName);
317     disp('db_2 loaded')
318 end
```

Appendix I.2

Script for cutting the velocity signal into single cycles

```
1  function [hw_v_cut,long,cut_nr] = cyc_cut(hw_v,rev_v,CAD_adj,N,SR,...
     CAD_switch)
2  %
3  % i=1
4  % hw_v = db_3.hw_v(:,i);
5  % rev_v = db_3.rev_v(:,i);
6  % CAD_adj = db_3.CAD_adj;
7  % N = db_3.rpm(i);
8  % SR = db_3.SR(i);
9
10 % Identifying peaks
11 lim = (max(rev_v)-min(rev_v))/2;
12 lim_index = rev_v < lim;
13
14 % Finding point for rising edge
15 cut_points = zeros(length(lim_index),1);
16 for i = 1:length(lim_index)
17     if lim_index(i) == 1 & lim_index(i-1) == 0
18         cut_points(i) = 1;
19     end
20 end
21
```

```
% Defining the positions of the rising edges
cuts = find(cut_points == 1);    % Compensate med CAD_adj
cuts_adj = cuts - ((60/(360*N))*CAD_adj*SR);
pos_index = cuts_adj > 0;
cuts_adj = cuts_adj(pos_index);
cut_nr = length(cuts_adj)-1;
% cuts_adj_plus = cuts_adj;

for i = 1:cut_nr-1
   cuts_adj_plus(i*2-1) = cuts_adj(i);
   cuts_adj_plus(i+1*i) = floor((cuts_adj(i+1)-cuts_adj(i))/2+cuts_adj...
     (i));
end
cut_nr = length(cuts_adj_plus)-1;

% Defining longest cut size
long = 0;
for i = 1:cut_nr;
    if length(hw_v(cuts_adj_plus(i):cuts_adj_plus(i+1),1)) > long
        long = length(hw_v(cuts_adj_plus(i):cuts_adj_plus(i+1)));
    end
end

% Cutting hw_v into cycles
cyc_hw_v_cut = zeros(long,length(cuts)-1);
count = 0;
for i = 1:length(cuts_adj_plus)-1
    if i/2 ~= round(i/2)
        count = count + 1;
    end
    stop = length(hw_v(cuts_adj_plus(i):cuts_adj_plus(i+1),1));
    cyc_hw_v_cut(1:stop,i) = hw_v(cuts_adj_plus(i):cuts_adj_plus(i+1))...
      ;
end

% Cutting hw_v into closed cycle - obtion of switcing 360CAD
hw_v_cut = zeros(long,400);
count = 0;
for i = 1:length(cuts_adj_plus)-1
    if count == 400
       break
    end
    if CAD_switch
        if i/2 ~= round(i/2)
            count = count + 1;
            hw_v_cut(:,count) = cyc_hw_v_cut(:,i);
        end
    else
        if i/2 == round(i/2)
            count = count + 1;
            hw_v_cut(:,count) = cyc_hw_v_cut(:,i);
        end
    end
end
cut_nr = count;



plotting = 0;
```

```
79
80  if plotting
81  figure(1)
82  subplot(4,1,1)
83  plot(hw_v)
84
85  subplot(4,1,2)
86  plot(rev_v)
87
88  subplot(4,1,3)
89  plot(lim_index)
90
91  subplot(4,1,4)
92  plot(cut_points)
93
94  figure(2)
95  for i = 1:length(cuts)-1
96     subplot(3,1,i)
97     plot(cyc_hw_v_cut(:,i))
98  end
99
100 figure(3)
101 for i = 1:cut_nr
102    subplot(cut_nr,1,i)
103    plot(hw_v_cut(:,i))
104 end
105 end
```

Appendix I.3

Script for simulating gas pressure and temperature

```
1   function [p,T,theta] = p_v_sim(N,increment,p_1,T_1)
2
3   % N = db_1.rpm(1);
4   % index = db_1.hw_v_cut(:,368,1) ≠ 0;
5   % increment = sum(index);
6   % p_1 = db_1.p_atm(1);
7   % T_1 = db_1.T_atm(1);
8
9
10  % Engine data
11  B = 65.08*10^(-3);      %[m]
12  S = 47.65*10^(-3);      %[m]
13  R = S/2;                %[m]
14  L = 88.90*10^(-3);      %[m]
15  epsilon = 6;            %[-]
16
17  % General data
18  gamma = 1.4;
19
20  % Volume and dV as function of CAD
21  theta = linspace(180,540,increment);      %[CAD]
22  Δ_theta = 720/increment;            %[CAD]
23
24  V_d = pi/4*B^2*S;           %[m^3]
```

```
25 V_c = V_d/(epsilon-1);      %[m^3]
26
27 V = V_c+(pi*B^2)/4*(R*(1-cosd(theta))+R^2/(2*L)*sind(theta).^2);     %[...
     m^3]
28 dV = (pi*B^2)/4*R*(sind(theta)+R/(2*L)*sind(2*theta));              %[...
     m^3/theta]
29
30 % ====== CREATING CORRECTION CURVE ======
31 N_meas = [500 1000 1500];     %[rpm]
32 p_meas = [2.75 7.5 8.25]*10^5;      %[Pa]
33
34 place = find(N_meas == N);
35 p_used = p_meas(place);
36
37 % ====== START OF SIMULATION ======
38
39 % Looking only on closed part of the cycle
40 %theta_index = theta > 180 & theta < 540 ;
41 %start = find(theta_index,1,'first');
42 %stop = find(theta_index,1,'last');
43
44 %CAD = theta(start:stop);
45 %Vol = V(start:stop);
46 %dVol = dV(start:stop);
47
48 % Initializing
49 p(1) = p_1;     %[Pa]
50 T(1) = T_1;     %[K]
51
52 % Simulating presssure
53 for i = 2:length(theta)
54     p(i) = p(1)*(V(1)/V(i))^gamma;
55 end
56
57 % Calculating correction factor
58 corr_factor = p_used/max(p);
59 p_corr = p*corr_factor;
60
61 % Calculating T form ideal gas law
62 T = p_corr.*V*T(1)./(p_corr(1)*V(1));
63
64 % ========= PLOTTING =========
65 correcting = 0;
66 results = 0;
67
68 if correcting
69 figure(1)
70 plot(N_meas,p_meas*10^(-5),'*')
71 title('Measured points')
72 xlabel('RPM [1/min]')
73 ylabel('Pressure [Bar]')
74 end
75
76 if results
77 figure(2)
78 subplot(2,2,1)
79 plot(theta,V,'b',theta,dV,'r')
80 title('Volume')
81 xlabel('CAD')
```

```
82  ylabel('[m^3]')
83
84  subplot(2,2,2)
85  plot(theta,p_corr*10^(-5),'b')
86  title('Pressure')
87  xlabel('CAD')
88  ylabel('[Bar]')
89
90  subplot(2,2,3)
91  plot(theta,T-273.15,'b')
92  title('Temperature')
93  xlabel('CAD')
94  ylabel('[C]')
95
96  subplot(2,2,4)
97  plot(V,p_corr*10^(-5),'b')
98  title('p-V diagram')
99  xlabel('Volume [m^3]')
100 ylabel('Pressure [Bar]')
101 end
```

Appendix I.4

Script for performing temperature correction

```
1  function [V] = T_corr(hw_v,p,T_g,cal_V,cal_E,nu_0,R_bridge,T_atm)
2  %
3  % index = hw_v_cut(:,1,1) ~= 0;
4  % hw_v = hw_v_cut(index,1,1);
5  % p = p(index,1,1);
6  % T_g = T(index,1,1);
7  % cal_V = db_0.cal_V(:,1);
8  % cal_E = db_0.cal_E(:,1);
9  % nu_0 = db_0.nu;
10 % R_bridge = db_0.R_bridge(1);
11 % T_atm = db_0.T_atm(1);
12
13 % Data
14 L_w = 2.2*10^(-3);  %[m]
15 D_w = 10*10^(-6);   %[m]
16 k_0 = 26.3*10^(-3); %[W/(m*K)]
17 a = 0.8;           %[-]
18 R_0 = 4.95;
19 alpha = 0.0016;     %[1/C]
20 R_spec = 287.058;   %[J/(kg*K)]
21 rho_0 = p(1)/(T_atm*R_spec);
22
23 % T_w from overheat ratio
24 T_w = (a/alpha+(T_atm-273.15))+273.15;       %[K]
25
26 % Calculating current wire resistance
27 R_w = R_0*(1+alpha*(T_w-T_atm));
28
29 % Calculating Nu from calibration voltage
30 Nu_cal = (cal_E/R_bridge).^2*R_w/(k_0*pi*L_w*(T_w-T_atm));
31
```

```
% Calculating Re from calibrating velosity
Re_cal = cal_V*D_w/nu_0;

% Calibration constants from calibration points
king_fit = fit(Re_cal,Nu_cal,'power2');
king_coef = coeffvalues(king_fit);
king_A = king_coef(3);
king_B = king_coef(1);
king_n = king_coef(2);

% Calculating V for each point
for i = 1:length(hw_v)
    k_cor = k_0*(T_g(i)/T_atm)^(0.8);

    Nu = (hw_v(i)/R_bridge)^2*R_w/(pi*L_w*k_cor*(T_w-T_g(i)));

    Re = exp(log((Nu-king_A)/(king_B))/king_n);

    mu_0 = nu_0*rho_0;
    mu_cor = mu_0*(T_g(i)/T_atm)^(0.76);
    rho = p(i)/(T_g(i)*R_spec);
    nu_cor = mu_cor/rho;
    V(i) = Re*nu_cor/D_w;
end
```

Appendix I.5

Script for cutting the cycles into intervals

```
function [V_int,int_CAD,int_length] = int_cut(V,CAD_int,CAD,int)

% V = db.V(index,1,1);
% CAD = db.CAD(index,1,1);
% CAD_int = db.CAD_int;
% int = db.interval;


int_length = ceil(length(CAD)/int);
V_int = zeros(int_length,int);
int_CAD = zeros(int_length,int);
for i = 1:int
    index = CAD > 180+CAD_int*(i-1) & CAD < 180+CAD_int*i;
    V_int(1:sum(index),i) = V(index);
    int_CAD(1:sum(index),i) = CAD(index);
end



% figure(22)
% for i = 1:int
% subplot(int,1,i)
% plot(int_CAD(:,i),V_int(:,i))
% end
```

Appendix I.6

Script for performing filtering This script is provided by Anders Ivarsson.

```
function filtsignal = fftBPfilter(time,signal,bandpasses,bw,plotflag)

% time:       Column vector were each element contain a time [s] that
%             correspond to elemnts in the signal vector. Also if the
%             signal is multidimensional array, the time has to a ... single
%             column vector that applies to all the signal columns.
%
% signal:     Column vector with even number of elements containing ... the
%             signal to be processed. The signal may also be two or ... three
%             dimensional arrays but then the filtering will be ... performed
%             columnwise.
%
% bandpasses: Matrix with 4 columns and a number of rows corresponding... to
%             the number bandpasses to be used. Each row defines a ... bandpass
%             by first and second element being the lower and upper ... band
%             frequencies [Hz]. Third and fourth element defines the
%             passfactors (between 0 and 1) at lower and upper band
%             frequencies. Passfactors between the band limits will be
%             interpolated linearly.
%
% bw:         Bandwidth [Hz] of the filter kernel as well as the ... bandwidth
%             of the transitions in the frequency respons. The ... transition
%             bandwidth will be symetrical around the specified ... bandpass
%             limit.
%
% plotflag:   Options: 'plot_on' or 'plot_off'. Determines whether or ... not
%             plots that show the filter performance will be ... generated.
%             The plot option will only display the filterperformance ... of
%             the signal in the first column if a multidimensional ... signal
%             array is used.

L = size(signal,1);
if round(L/2) ~= L/2
    error('The signal must have an even number of elements')
end
fmax = (L/2)/diff(time([1,end]));
M = 2*round(2/(bw/(2*fmax)));
if M > L
    M = L;
```

```
40 end
41 f = [0:L/2-1]'/diff(time([1,end]));
42
43 KERNEL = zeros(L/2,1);
44 for i = 1:length(bandpasses(:,1))
45     index = f ≥ bandpasses(i,1) & f < bandpasses(i,2);
46     KERNEL(index) = interp1([1 find(index,1,'last')]',bandpasses(i...
         ,3:4)',find(index)','linear');
47 end
48 KERNEL = [KERNEL; KERNEL(end:-1:1)];
49 KERNEL = KERNEL + j*KERNEL;
50
51 kernel = real(ifft(KERNEL));
52 black = blackman(M);
53 black = [black(M/2+1:end); zeros(L-M,1); black(1:M/2)];
54 kernel = kernel.*black;
55
56 KERNEL = fft(kernel);
57
58 if 0
59 kernel2 = ifft(KERNEL.^2);
60 KERNEL2 = fft(kernel2);
61 kernel = kernel2;
62 end
63
64 KERNELlong = fft([kernel(1:L/2); zeros(2*M,1); kernel(L/2+1:end)]);
65 KERNELlong = repmat(KERNELlong,[1,size(signal,2),size(signal,3)]);
66 signallong = [repmat(signal(1,:,:),[M,1,1]); signal; repmat(signal(L...
     ,:,:),[M,1,1])];
67 i_signal = logical([zeros(M,size(signal,2),size(signal,3)); ones(size(...
     signal)); zeros(M,size(signal,2),size(signal,3))]);
68
69
70 fftsignallong = fft(signallong);
71 filtsignallong = ifft(fftsignallong.*KERNELlong);
72 filtsignal = zeros(size(signal));
73 filtsignal(:) = filtsignallong(i_signal);
74
75 if strcmp(plotflag,'plot_on')
76
77     figure
78     plot(f,abs(KERNEL(1:L/2)))
79 %   plot(f,abs(KERNEL(1:L/2)),'-b',f,abs(KERNEL2(1:L/2)),'-r')
80     title('Frequency response of filter kernel')
81     xlabel('Frequency [Hz]')
82     ylabel('Normalized amplitude')
83
84     before = 2*abs(fft(signal(1:L))/L);
85     after = 2*abs(fft(filtsignal(1:L))/L);
86     figure
87     plot(f,before(1:L/2),'-k',f,after(1:L/2),'-r')
88     legend({'Unfiltered','Filtered'})
89     title('Single sided amplitude spectrum of signal before and after ...
         filtering')
90     xlabel('Frequency [Hz]')
91     ylabel('Amplitude [Same units as ingoing signal]')
92     xlim([0 f(end)])
93     maxy = 10*max(mean(before),mean(after));
94     %ylim([-0.01*maxy 1.2*maxy])
```

```
95      ylim([-0.01 1])
96
97      figure
98      plot(time,signal(1:L),'-b',time,filtsignal(1:L),'-r')
99      title('Signal in time domain before and after filtering')
100     legend({'Unfiltered','Filtered'})
101     xlabel('Time [s]')
102     ylabel('Signal units')
103
104 end
```

Appendix J

MatLab script for collecting data bases and plotting

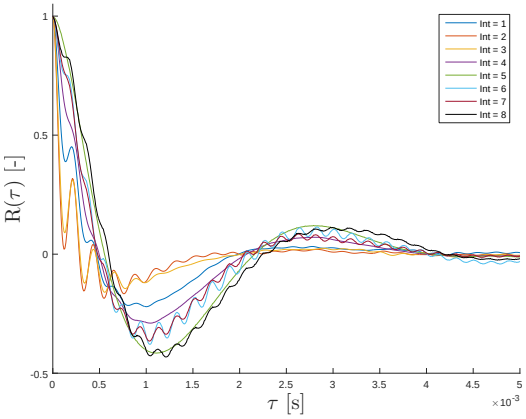
```
1
2  loading = 1;
3
4  if loading
5     clear all
6     loading = 1;
7  end
8
9  close all
10 clc
11
12 standard_plots = 1;
13 scales_plots = 1;
14 autocorrelation = 1;
15 energy_plots  = 1;
16
17 % ===== nomeclatur =====
18 %   1 ¬ 500rpm
19 %   2 ¬ 1000rpm
20 %   3 ¬ 1500rpm
21 %   4 ¬ 1000rpm – sweap
22
23 if loading
24 % Loading data-bases
25 load('db_1.mat'); disp('db_1 loaded');
26 load('db_2.mat'); disp('db_2 loaded');
27 load('db_3.mat'); disp('db_3 loaded');
28 %load('db_4.mat'); disp('db_4 loaded');
29 end
30
31 my_colormap = [0 0.4470 0.7410; 0.8500 0.3250 0.0980; 0.9290 0.6940 0...
     .1250;...
32                0.4940 0.1840 0.5560; 0.4660 0.6740 0.1880; 0.3010 0.7450...
                     0.9330;...
33                 0.6350 0.0780 0.1840; 0 0 0];
34 x_int = [1:8]';
35
36 if standard_plots
37     V_avg_plots(db_1,db_2,db_3,my_colormap,x_int)
38     RTI_plots(db_1,db_2,db_3,my_colormap,x_int)
39 end
40 if scales_plots
41     taylor_microscale(db_1,db_2,db_3,my_colormap,x_int)
42      intergral_time_and_length_scales_plots(db_1,db_2,db_3,my_colormap...
          ,x_int)
43      kolmogorov_time_and_length_scales_plots(db_1,db_2,db_3,...
          my_colormap,x_int)
44 end
45 if autocorrelation
46    autocorrelation_plots(db_1,db_2,db_3,my_colormap)
47 end
48 if energy_plots
49     energy_spectrum_plots(db_1,db_2,db_3,my_colormap)
```

```
50  end
```

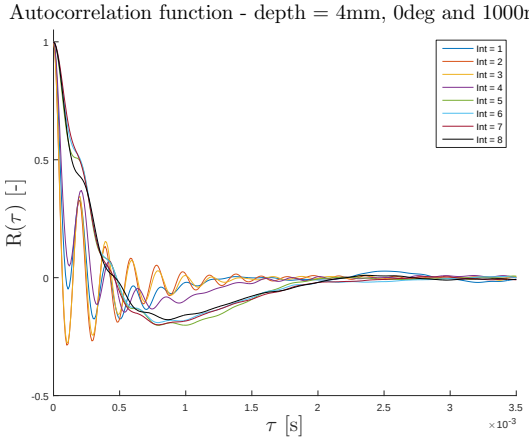
Appendix K

All autocorrelation functions at different positions and engine speeds

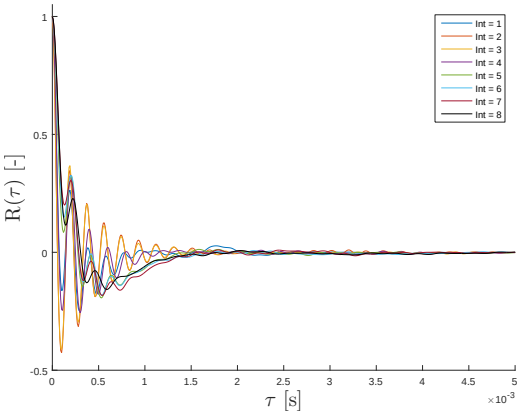
Autocorrelation function - depth = 4mm, 0deg and 500rpm



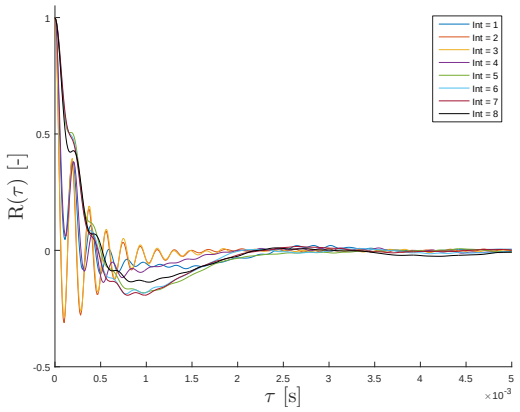
Autocorrelation function - depth = 4mm, 0deg and 1000rpm



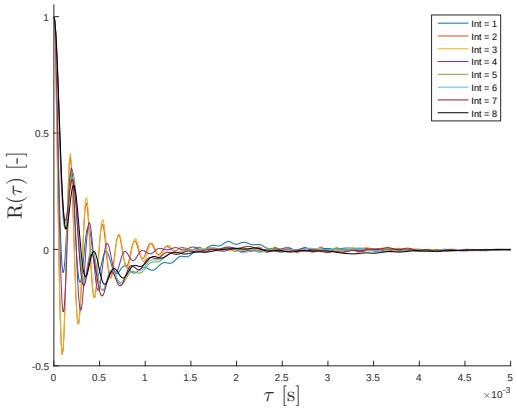
Autocorrelation function - depth = 4mm, 0deg and 1500rpm



Autocorrelation function - depth = 4mm, 90deg and 1000rpm



Autocorrelation function - depth = 4mm, 90deg and 1500rpm



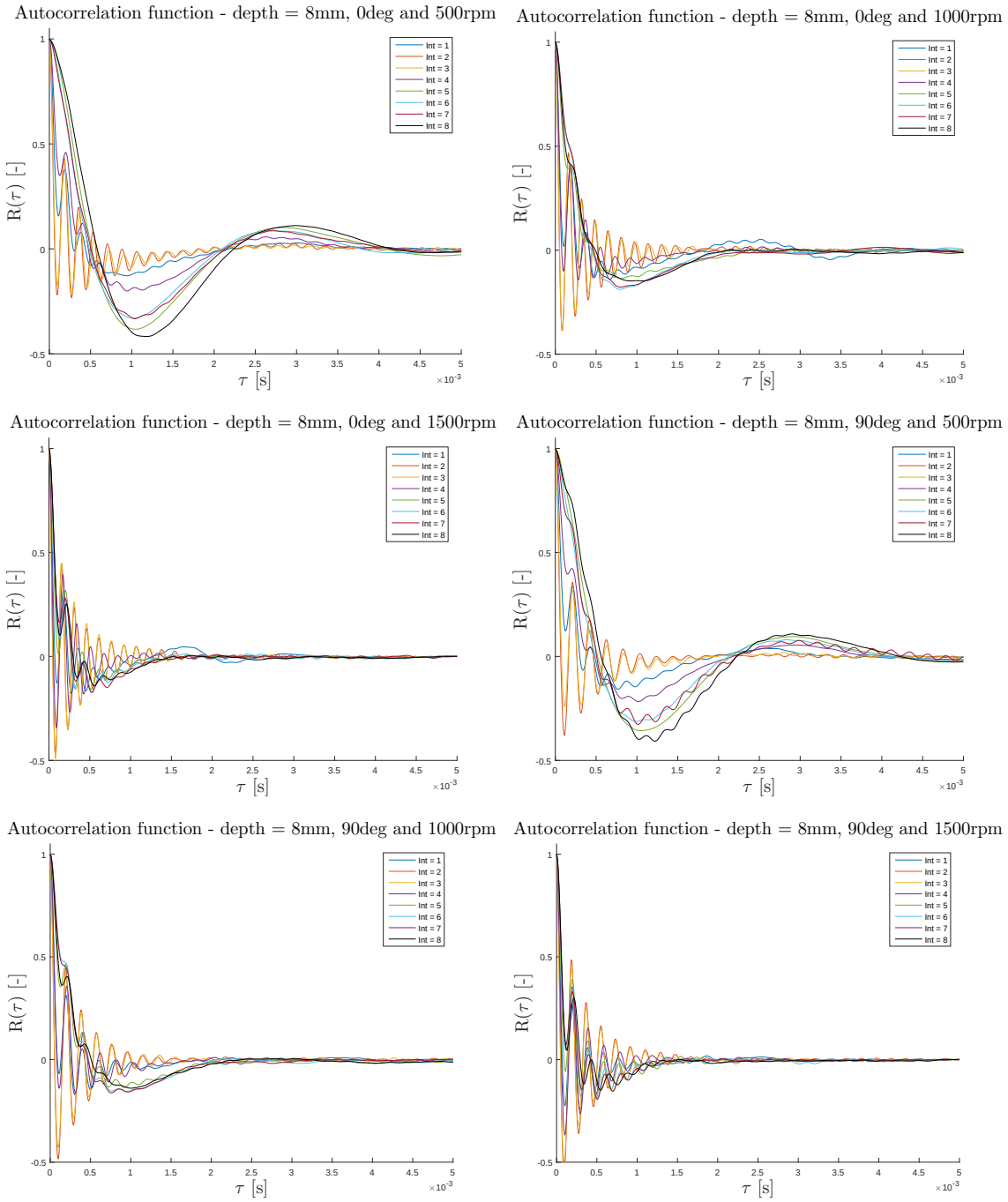
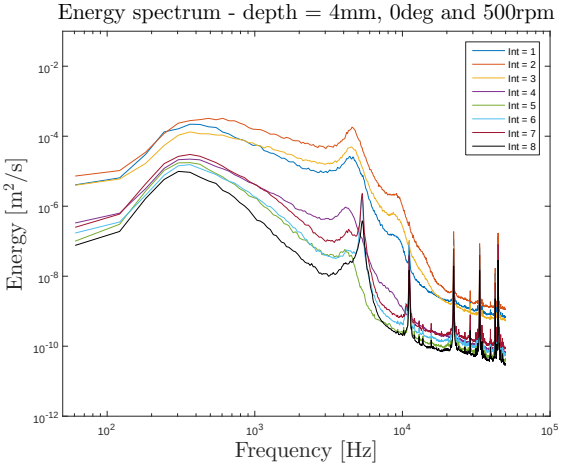


Figure 70: Autocoorelation function at the different positions and engine speeds

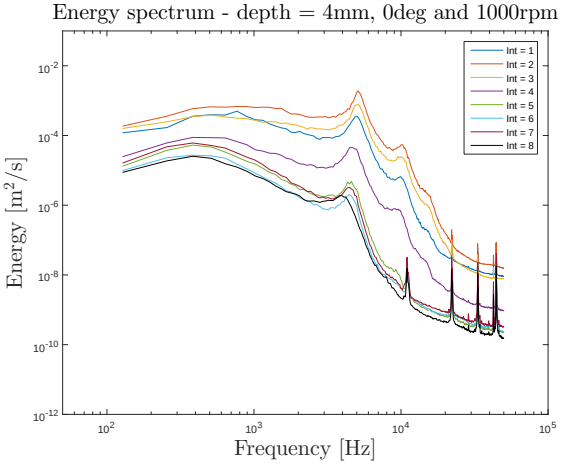
Appendix L

All energy spectra at different positions and engine speeds

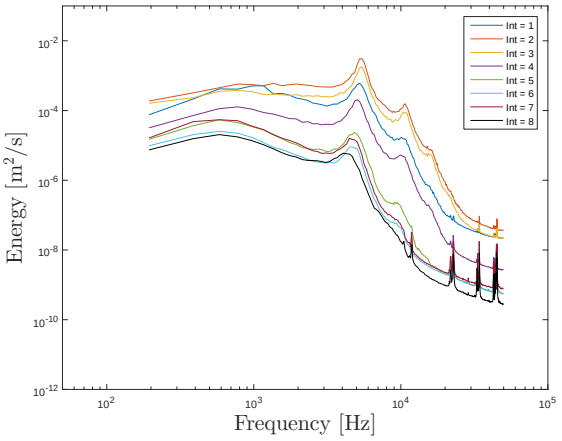
Energy spectrum - depth = 4mm, 0deg and 500rpm



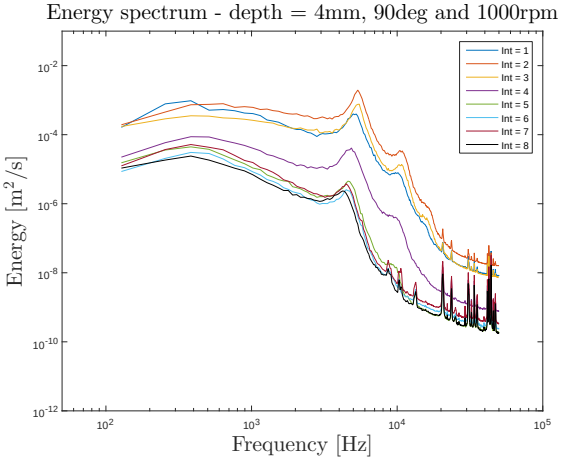
Energy spectrum - depth = 4mm, 0deg and 1000rpm



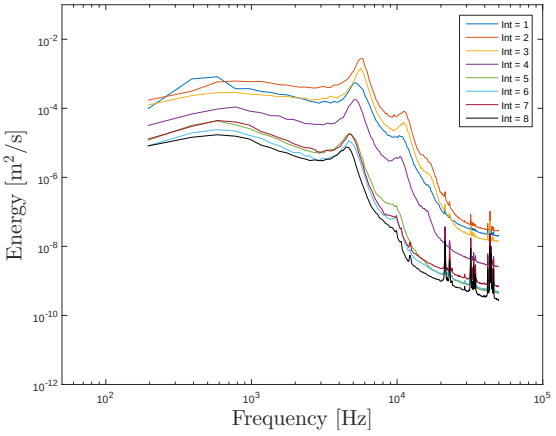
Energy spectrum - depth = 4mm, 0deg and 1500rpm



Energy spectrum - depth = 4mm, 90deg and 1000rpm



Energy spectrum - depth = 4mm, 90deg and 1500rpm



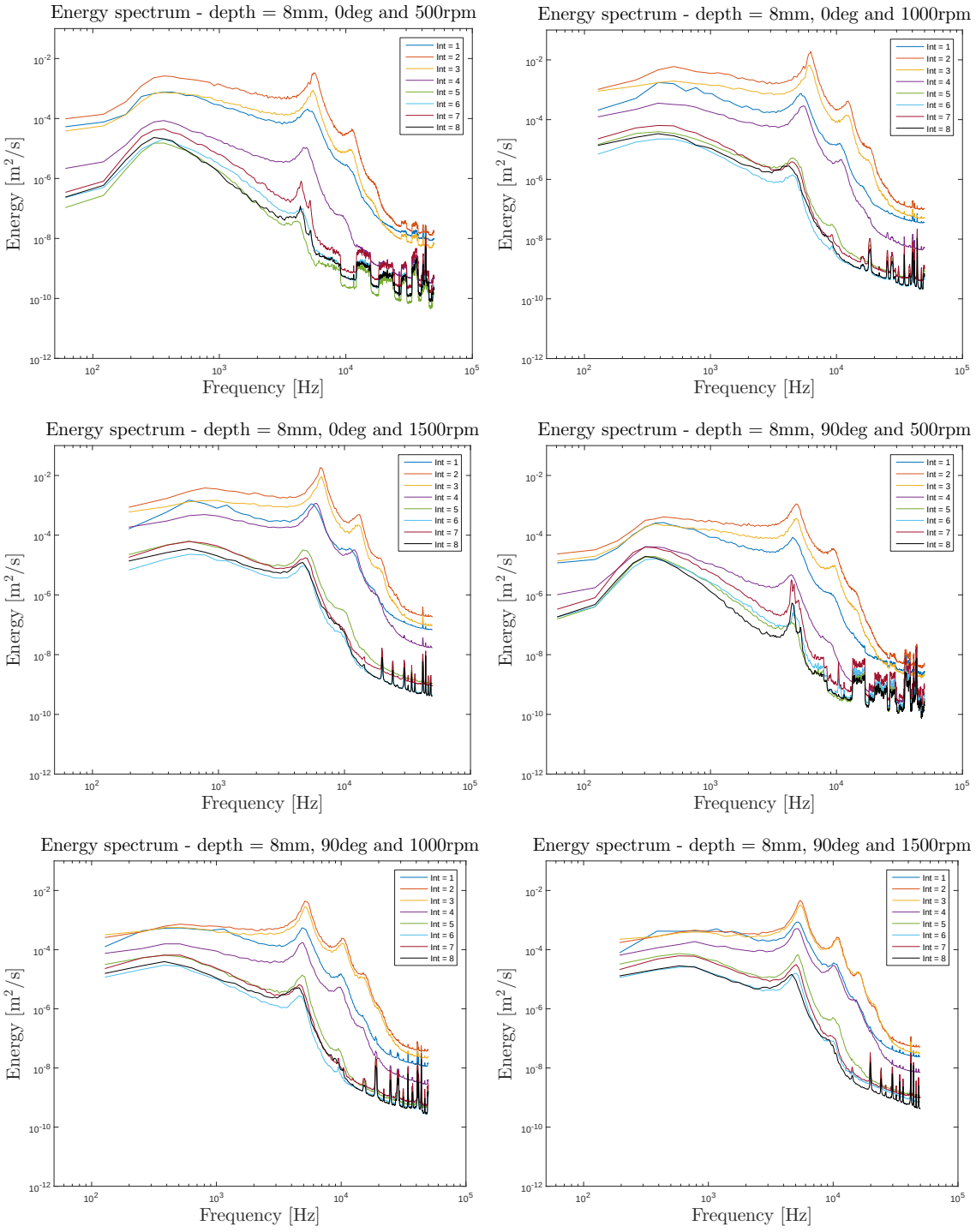


Figure 71: Energy spectra at the different positions and engine speeds