Implementation of an Automated Position and Tension Determination of Wires in Multi-Wire Proportional Chambers

Murat Esen

2018

Vorgelegt an der Goethe-Universität Frankfurt am Main
Institut für Kernphysik

Implementation of an Automated Position and Tension Determination of Wires in Multi-Wire Proportional Chambers

Bachelorarbeit
von
Murat Esen
Frankfurt am Main
15. November 2018

Contents

1 Introduction

Relativistic heavy-ion physics investigates dense and hot nuclear matter. Densities are examined, which are many times higher than those of normal nuclear matter and temperatures, that are one hundred thousand times higher than the temperature in the center of the sun. Under these extreme conditions a phase transition of matter to the so-called quark-gluon plasma (QGP) is expected. Fig. 1.1 shows the theoretically predicted phases of quantum chromodynamics (QCD).

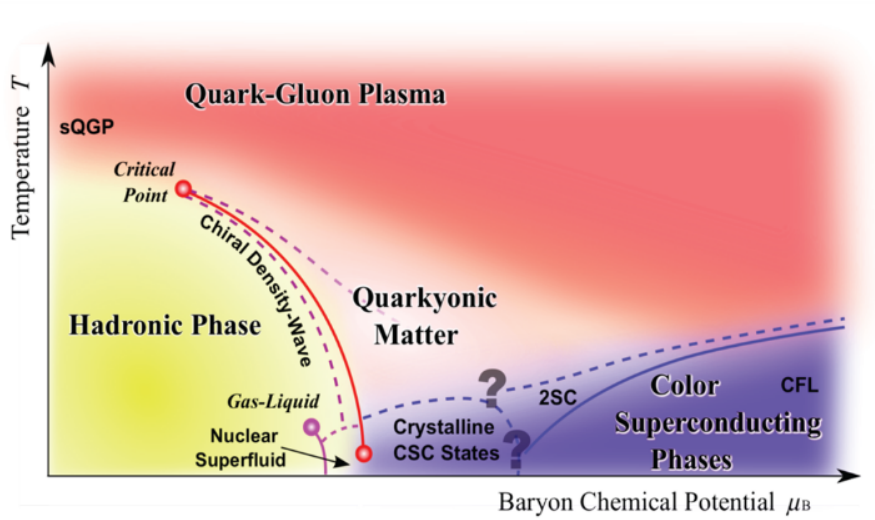

Figure 1.1: Sketch of the phase diagram for strongly interacting matter [5].

In order to produce such a condition in the laboratory, accelerator systems collide atomic nuclei with high energy. In these collisions many particles are generated in the region where the nuclei pass through each other - also called "fireball". The hadrons interact with their surrounding particles before leaving this "fireball"-region. Some of them are unstable and decay before reaching the detectors. But the generated leptons however pass through this "fireball" unimpeded, since they are not influenced by the strong nuclear interaction. Therefore leptons are good probes that can be used to make statements about the state of matter in the early phase of the reaction from which they originated. Produced leptons and hadrons, mainly pions, are leaving very similar signals in most measuring instruments.

In order to be able to differentiate between hadrons and leptons a special detector has been developed, the so-called *Transition Radiation Detector* (TRD). It exploits the fact that very fast ultrarelativistic electrons emit *Transition Radiation* (TR) in the X-ray energy range, when they pass through material layers with different plasma frequencies. A readout chamber behind such a radiator is used to detect this radiation. In the gas-filled chamber electrons, which are generated by the incoming particle in the gas due to ionization, drift in an electrical field to the direction of a readout electrode at which they can be detected. In order to produce a constant electromagnetic field and thus also constant gas gain over the entire chamber, two layers of thin wires are used. During the production of these readout chambers it must be ensured that the values for tension and position of all wires fulfill a certain accuracy. Otherwise, an inhomogeneous electric field would develop in the chambers and the gas gain and thus the measured data would deteriorate. Since a numerous number of such chambers will be built and each equipped with up to 400 wires, a manual check of the tension and position of each wire could hardly be carried out. For this reason a machine was developed, the so-called *Wire-Test-Device* (WTD) [4]. The purpose of this device was to carry out the tension and position measurement automatically. The improvement of this device, especially the newly written program code of the software, is the content of this thesis.

2 Physics Motivation

2.1 The Standard Model of Particle Physics

Physicists have summarized their knowledge of the smallest particles in the so-called standard model of particle physics. It perfectly describes all the phenomena of the microcosm known to us and includes the following types of elementary particles: six quark types, six leptonic states, the respective antiparticles to these twelve states and thirteen bosonic states. The elementary particles in the standard model are divided in four different classes: the quarks, the leptons, the gauge bosons (the exchange particles of forces) and the Higgs boson. Quarks and leptons are fermions[1], the exchange particles are vector bosons[2] and the Higgs is a scalar boson[3]. Fig. 2.1 shows the building blocks of the standard model. In addition, quarks and leptons are divided in three generations. The division into these generations is marked with I, II and III.
In the standard model, the concept of exchange particles is used as a virtual transmitter of the force effect. The elemental forces are the strong, the electromagnetic, the weak and the gravitational interactions. However the gravitation is not considered in the standard model. The masses of the gauge bosons are related to the ranges of the interaction and have been experimentally determined. The quarks and gluons are subject to the strong interaction, their bound states are called hadrons. Hadrons themselves are divided into two groups, the baryons and the mesons. The assignment is made via the baryon number B of Hadrons. Quarks are $B = \frac{1}{3}$, antiquarks $B = -\frac{1}{3}$. All hadrons with $B = \pm 1$ are assigned to the baryons and all hadrons with $B = 0$ are assigned to the mesons [19].

[1] The term "fermion" describes all particles with half-integer spin.
[2] The term "vector boson" describes all particles with integer spin equal to 1.
[3] The term "scalar boson" describes all particles with spin 0.

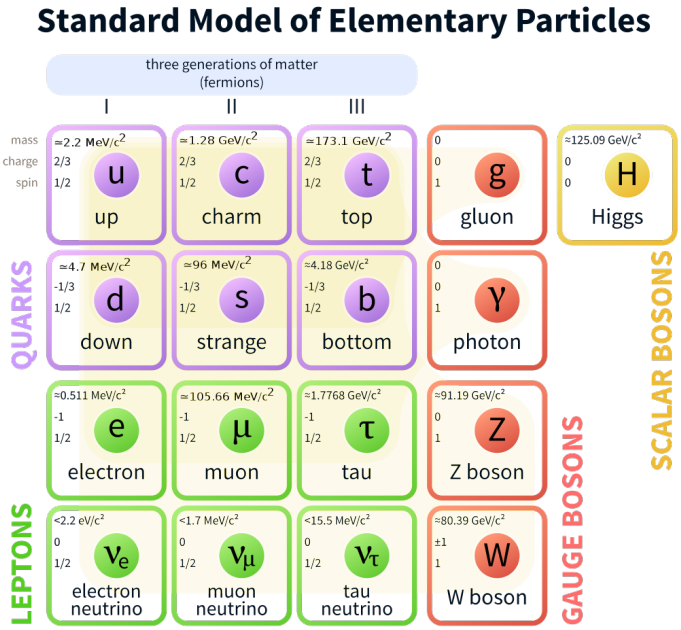

Figure 2.1: The standard model of particle physics showing the different generations of quarks and leptons as well as the exchange particles of the four fundamental interactions and the Higgs boson [20].

2.2 Compressed Baryonic Matter Experiment

The *Compressed Baryonic Matter* experiment (CBM) is one of the main experimental setups of the *Facility for Antiproton and Ion Research* (FAIR) project. FAIR is an international accelerator facility presently under construction and will be combined with the existing facilities of the *Helmholtz Centre for Heavy Ion Research* (GSI) located near Darmstadt, Germany (see Fig. 2.2). At FAIR research in the fields of nuclear, hadron and particle physics, atomic and anti-matter physics, high density plasma physics, and applications in condensed matter physics, biology and bio-medical sciences will be performed under usage of antiprotons and ions collisions [17].

The four scientific pillars of FAIR are the following:

- **C**ompressed **B**aryonic **M**atter (CBM)
- **A**tomic, **P**lasma **P**hysics and **A**pplications (APPA)
- **NU**clear **St**ructure, **A**strophysics and **R**eactions (NUSTAR)
- anti**P**roton **AN**nihilation at **DA**rmstadt (PANDA)

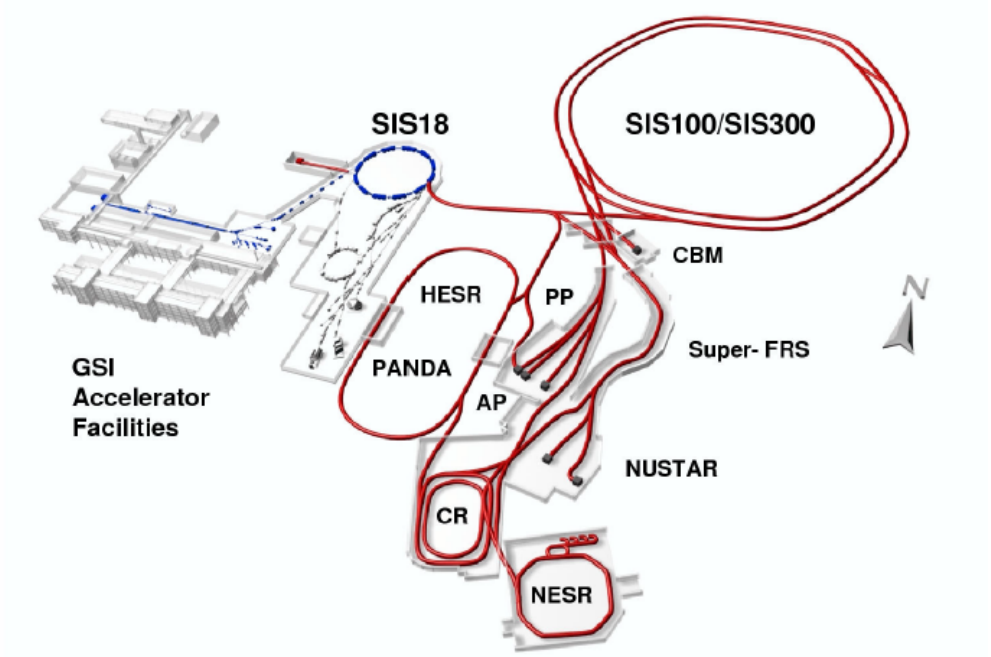

Figure 2.2: Overview of the future FAIR complex [5].

The objective of the CBM research program in particular is the examination of the phase diagram of QCD using high-energy nucleus-nucleus collisions, especially in the region of extreme baryon densities, i.e. at a high baryochemical potential. This research field can be subdivided in the study of the nuclear matter equation-of-state, the understanding of neutron star matter, the investigation of certain phase transition types and orders, chiral symmetry restoration and exotic forms of QCD matter (strangeness, etc.). One of the most important CBM experiment extensions comes along with a new particle accelerator consisting two extension stages.
The CBM detector is designed to conduct high precision measurements of the collective behavior of hadrons and rare diagnostic probes such as multi-strange hyperons, charmed particles and vector mesons (e.g. the ω, ρ and ϕ) decaying into lepton pairs, while also providing extensive statistics. In order to achieve those required statistics, the detector design and electronics need to be able to handle high interaction rates and therefore have to be fast.
Also the modular design of the CBM experiment will allow two separate measurement setups. One being a system specially designed for the identification of muons and one which is optimized for the measurement of electrons and hadrons (see Fig. 2.3). To change between these two versions, individual subsystems are meant to be exchangeable [5].

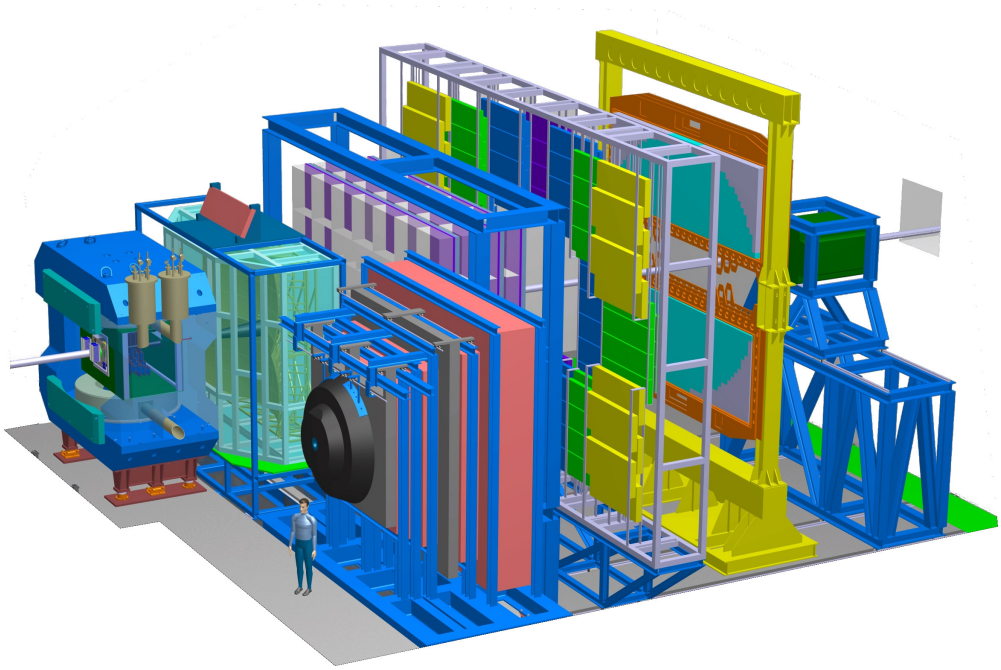

Figure 2.3: Sketch of the CBM detector [21].

Right behind the collision point are the *Micro Vertex Detector* (MVD) and the *Silicon Tracking System* (STS). These two CBM detector subsystems are placed in the interior of a dipole magnet and are used for vertex and track reconstruction. From the curvature of the track the momenta of the particles are determined.
The dipole magnet is followed by one of two exchangeable modules. One module is a so-called *Ring-Imaging-Cherenkov-Detector* (RICH), which is used to identify electrons and positrons in the momentum range of about $0-6$ GeV/c. In order to measure muons, the RICH detector is exchanged with *Muon Chambers* (MuCh), which are several absorber layers of iron plates combined with tracking chambers.
This interchangeable module is followed by a TRD, which is installed to identify electrons with momenta above 1.5 GeV/c and therefore extends the capability to discriminate pions from electrons beyond the range of about the RICH detector. Furthermore, a determination of the charge of fragments to separate e.g. deuterium- and helium-ions is possible by measuring their energy loss in the TRD.
Following the TRD is a wall of resistive plate chambers for the *Time-Of-Flight* (TOF) measurement. The TOF measurement allows the identification of charged hadrons via the determination of their velocity. An *electromagnetic calorimeter* (ECAL) is used for the measurement of photons and electrons. The *Projectile Spectator Detector* (PSD) at small forward-angles is used in the electron setup to measure the centrality of the collisions [5].

In summary, the entire CBM detector setup faces the challenge of handling high event rates as well as high particle rates within a single event itself. Here a continuous readout procedure without data-event assignment is required, since trigger based readout procedures would be too slow in handling the planned event rates.
The actual assignment of the data stream from subsystems to events takes place afterwards via an online event-reconstruction providing a global time stamp. It is crucial that subsystems themselves must be developed for processing high event rates and particle hit rates. Also the readout data must be formatted in such way that the data volume remains manageable [5].

2.3 Multi-Wire Proportional Chamber

The principle of a *Multi-Wire Proportional Chamber* (MWPC) primarily consists of a layer of wires at high voltage (anode) running through a chamber with two conductive walls held at ground potential (cathode). The purpose of a MWPC is the detection of charged particles and photons and it thus provides spatial information on the particle trajectory by tracking the trails of gaseous ionization. In combination with the walls, these wires generate a radial electric field drawing extra electrons or negative ions to the anode wires with negligible lateral motion (see Fig. 2.4).
Usually these chambers are filled with a noble gas mixture, e.g. xenon and argon. An ionizing particle passing through the chamber ionizes the gas atoms. The ions and electrons resulting from this ionization process are accelerated by the electric field across the chamber, causing a local electron avalanche on the anode wires since in the immediate vicinity of the wires the electric field strength is particularly high.
The electron avalanche effect is proportional to the ionization effect of the detected particle. The multiplication of ionization electrons is called *gas amplification* or *gas gain* and depends on the potential of the anode wires. These informations allow the reconstruction of the total energy deposition inside the chamber.
More sophisticated MWPCs have an additional cathode wire plane that divides the chamber into two parts, a pure drift part and an amplification part. The amplification part is not limited by a simple conductive wall. Instead a so-called pad plane consisting of several small pads is used and the induced signal is collected by these pads. Due to this pad arrangement, a two-dimensional spatial resolution can be obtained.
MWPCs are an option to be considered for experiments like the CBM since they make large detector setups affordable and particles pass through them without being exposed to too many other sources of influence [1].

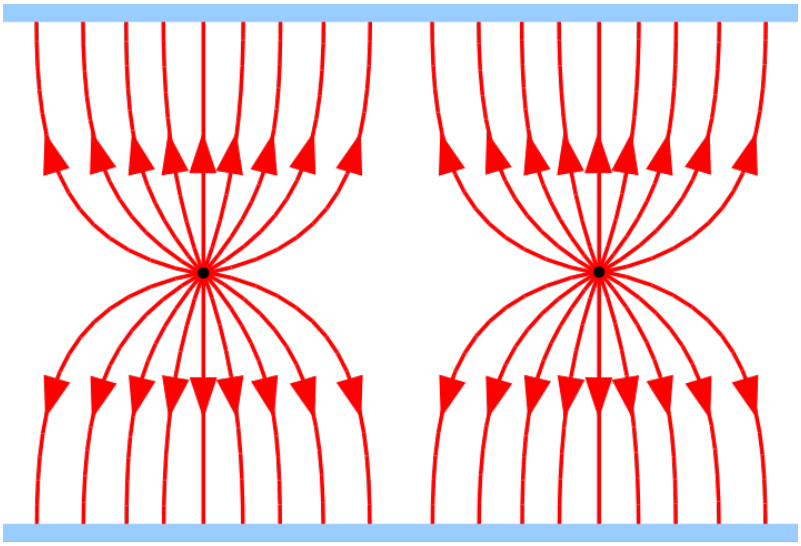

Figure 2.4: The electric field lines in a simple MWPC. The blue walls form the cathodes and are held at ground potential [16].

2.4 Transition Radiation Detector

To distinguish pions from electrons a detector for transition radiation photons (TR-photons) is required. The TRD basically consists of two parts, the readout chamber (ROC) and the radiator. Since the production of transition radiation depends on the Lorentz factor γ of a particle, TR-production differs significantly for high and low mass particles. The latter reach much higher γ-values in comparison, allowing low mass particles such as the electron to be detected by their TR, while the same is suppressed for heavy particles such as pions.
In order to efficiently absorb the TR-photons in the gas volume of the ROC, a xenon counting gas mixture is used, having a high absorption cross section for photons in the TR spectral range. This ensures the absorption of TR-photons predominantly in the region directly behind the entrance window.
The yield of TR-photons can be calculated directly as a function of the Lorentz factor γ for a regular radiator consisting of foil layers with fixed distances, assuming the number of foils N_f, their thickness l_1 and distance l_2 is given[4]. Thus, a regular radiators advantage is given by the fact that it can be analytically calculated. Alternatively, there are irregular radiators made of foams. These have the disadvantage that the analytical calculability of TR is not given due to varying air bubbles within these foams. Their advantage lies in their feasibility in financial and technical terms. However, materials being suitable for a given setup need to be found firstly [1].

[4]Here, we additionally assume $l_1 < l_2$

The CBM-TRD consists out of four detector layers. It is composed of the radiator and the Read-Out Chamber (ROC). The ROCs are designed as MWPCs with an amplification region of 3.5+3.5 mm thickness and a drift region of 5 mm. Their implementation in the simulation framework can be seen in Fig. 2.5 [1].

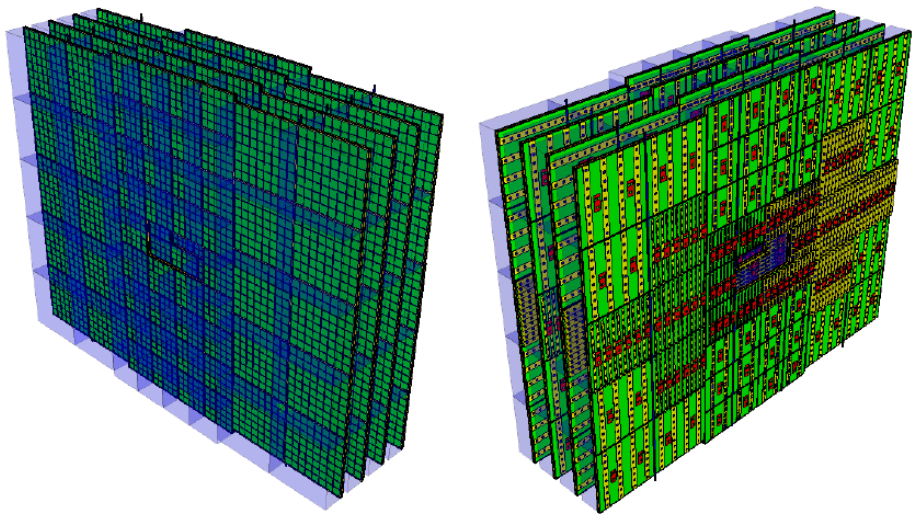

Figure 2.5: Shown here is the implementation of the TRD geometry in the CBMROOT simulation framework. Visible are the ROCs with the radiator boxes in the front view (left), while the rear view (right) shows the back panels of the ROCs together with the front-end electronics [5].

3 Wire-Test-Device

3.1 Purpose

The electric field of a TRD readout chamber depends on the quality and position of the anode and cathode wires. The electric field around a wire depends on its diameter. Thinner wires can produce higher field densities around them, resulting in stronger field gradients [15]:

$$E_r = \frac{1}{2\pi\epsilon_0}\frac{\lambda}{r} \tag{3.1}$$

In this case $\lambda = \frac{Q}{L}$ is the linear charge density, L is the wire length, r is the radius and $\epsilon_0 = 8.854 \cdot 10^{-12}$ $\mathrm{A^2s^2/Nm^2}$ is the dielectric constant. The anode wires are made of gold-plated tungsten (AuW[1]) with a density of 19.29 $\mathrm{g/cm^3}$ [3] and a diameter of 20 µm [3]. The cathode wires consist of an alloy of copper and beryllium (CuBe[2]) with a density of 8.23 $\mathrm{g/cm^3}$ [3] and a diameter of 75 µm [3].
The multiplication characteristics of a gas chamber, as well as the drift behavior of the electron cluster in the chamber, depend on the condition of the electric field. Therefore, the electric field generating wires need to fulfill requirements to produce an uniform electric field and therefore constant drift times over the whole chamber. It is also necessary that the distances between the wires must be equal so that missing or shifted wires need to be detected early. Furthermore, it is important to check the tension of the wires on deviations from the standard value. These procedures are necessary, so that the electric field can remain constant to the middle of the chamber. A broken anode wire can even cause a short circuit with the cathodes. In order to hold the anode wires in position, they are glued to the chamber with a nominal tension of approximately 0.50 - 0.55 N. The thicker cathode wires are stretched with approximately 1 N.
For quality control the wires used in the TRD are tested for tension and position on the chamber. In previous experiments the wire tension was checked with a manually operated device. In this method, each wire had to be touched with a pen-shaped instrument. In this case damage to the wire or the change of wire tension could not be ruled out. The wire positions were not measured at this time. Another disadvantage of this method was the required time: the measurement of a single wire lasted about 30 seconds. For a TRD chamber with up to 400 wires, it would take over 3 hours [4].

[1] Au stands for gold and W for tungsten, according to the periodic table of the elements.
[2] Cu stands for copper and Be for beryllium, according to the periodic table of the elements.

The main goal was to develop a method to gently measure the tension and position of the wires. Therefore, a contactless optical measurement has been proposed and built. The WTD has been developed to measure automatically the tension and position of all anode and cathode wires of a TRD drift chamber without contact (see Fig. 3.1). For one chamber, the WTD needs about 15 minutes. Apart from the temporal advantages, the results of the measurements were many times more accurate and reliable than in manual measurements [4].

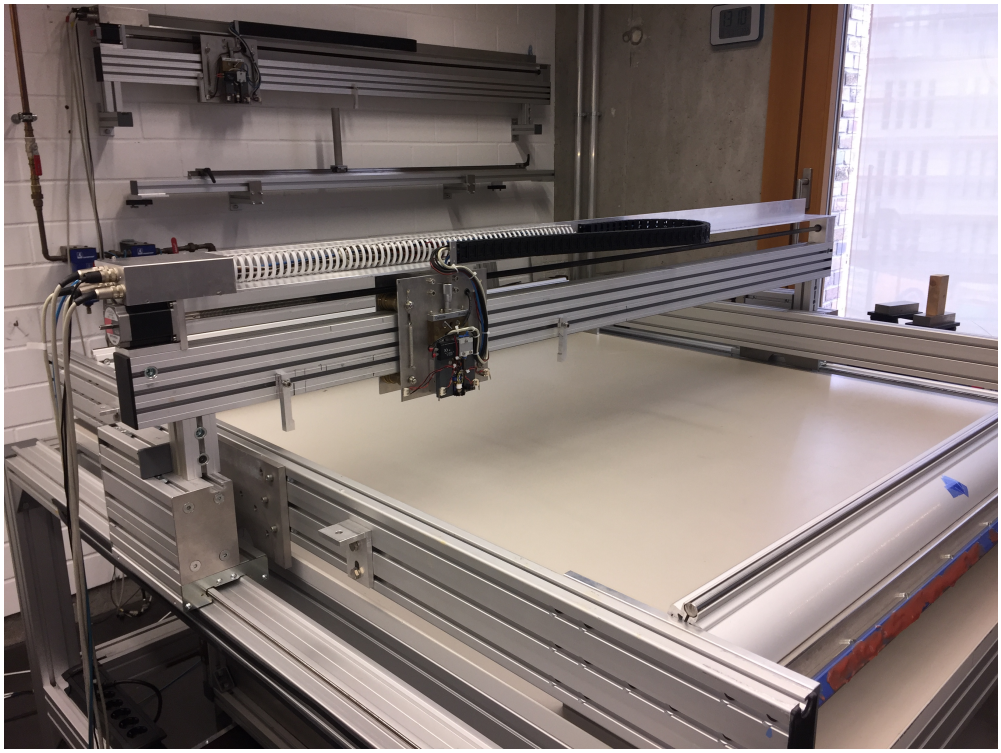

Figure 3.1: Photo of the WTD measuring the wires of a frame winding.

3.2 Setup

The following section describes in more detail the mechanical and electronic implementation of the components. The machine consists of a support framework, made of a 2 m long aluminum profile and a carriage which can be moved over it, on which the measuring instruments are mounted.
The framework structure of the carriage consists of two aluminum plates. Four brass rollers, two above and two below, stabilize the movement of the carriage. The two rollers below the running profile are fixed, the two above are spring-loaded to compensate for any unevenness on the profile.
Behind the aluminum profile a magnetic length measuring system is attached, the so-called *POSIMAG* from ASM Sensors. POSIMAG is used to measure wire positions. A magnetic sensor attached to the measuring head is moved over a magnetic tape provided with a sinusoidal magnetization and reads it out. There are two readout channels in the sensor which are phase shifted by 90° in terms of the magnetization on the tape. An electronics in the sensor generates a TTL[3] square wave signal from the read sine curve. When the signal threshold is reached, the state changes between TTL-high and TTL-low. Fig. 3.2 shows the phase-shifted output signal of the two channels of the sensor head. The two signals are combined in an external quadrature decoder so that the lower "signal train" seen in figure 3.2 is created. This achieves a quadrupling of the spatial resolution. If the voltage amplitudes of the photo sensor are recorded together with position information from the quadrature decoder, the position of the wires can be determined by looking for the maxima in the photo sensor amplitude spectrum [4].

[3] **T**ransistor **T**ransistor **L**ogic, output is a square-wave voltage with two possible states: a logical 1 corresponds to a 5 V signal, a logical 0 to a 0 V signal.

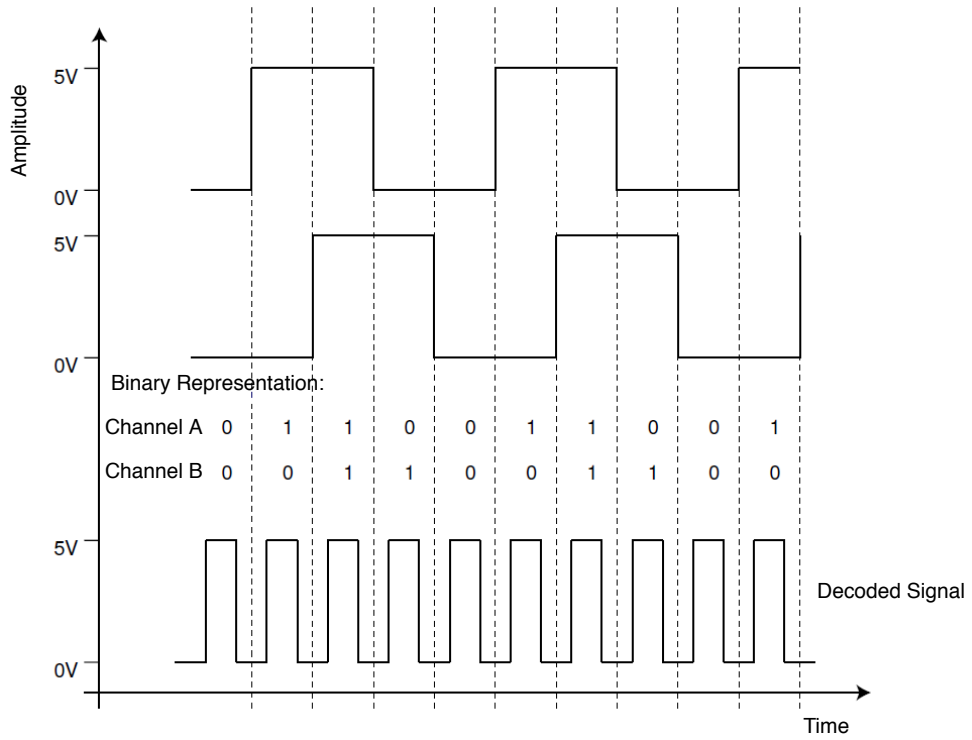

Figure 3.2: Principle of position determination with the magnetic length measuring system POSIMAG from ASM Sensors. Each pulse of the decoded signal corresponds to a particular traveled path [4].

On the carriage the optical sensor with a preamplifier, the magnet valve and, on the back, the readout head of the POSIMAG length measuring system are mounted (see Fig. 3.3). In addition, there are four limit switches, two on the aluminum baseplate and two on the optical sensor (two each for the two directions of motion). Since the focus of the optical sensor must be adjusted to the wire plane to be measured, the preamplifier and sensor are mounted onto an adjustable sample carrier. The focus can be precisely adjusted with a micrometer screw. For fast selection of anode and cathode wire planes, the measuring head can be lowered and lifted by 3.5 mm with a small lever [4].

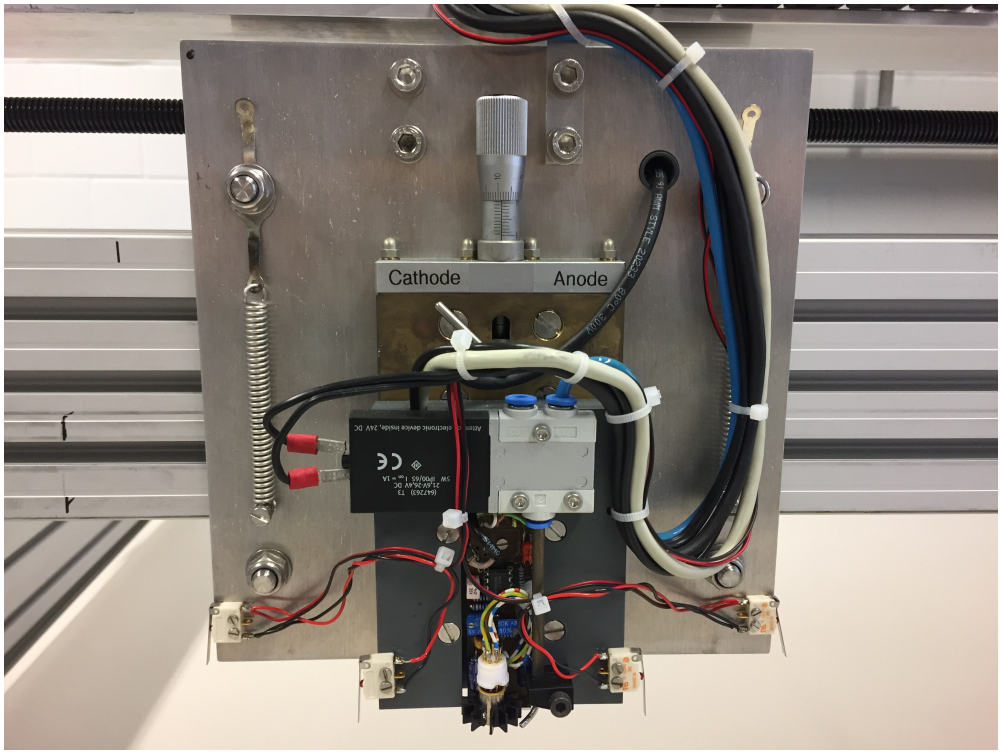

Figure 3.3: Closeup of the probe carrier. The optical sensor with preamplifier, the solenoid valve with nozzle and the adjustment screw can be seen.

The cable management for the measuring instruments on the transport carriage takes place via a drag chain above the driving beam. On the motor side of the system, sockets are mounted for all electrical connections. The pressurized gas is fed from a nitrogen cylinder (nitrogen 5.0[4]) via a 4 mm thick hose to the machine.
In the wire test device, a hybrid stepper motor from Oriental Motors with a maximum current of 2.8 A per phase was used. Stepper motors allow to move to certain paths exactly. For example, stepper motors are used in printers or Bluray players. By switching coils on and off, the rotor can be rotated by a certain angle. Single TTL pulses which are handed over to a control electronics make the rotation of the rotor by small angle controllable.
The entire electronics is installed in a 19 inch case (see Fig. 3.4). To carry out repair and maintenance work quickly and conveniently, the electronics has a modular design. Each module can be removed individually and quickly replaced [4].

[4]Nitrogen with a purity of 99.999 vol.% is used. It is important that there are no traces of contamination. Permanently installed compressed air lines may contain traces of oil that can contaminate the TRD chambers.

All components are interconnected via a common bus system on which all necessary control and data signals can be transmitted.

Figure 3.4: Photo of the electronics. On the front are the connections for the WTD and on the back the interface to the PC and the power supply.

Power supply and control: The power supply provides four different voltages: +24 V, +12 V, +5 V and -12 V. They are connected to the bus via a microfuse. A power failure is indicated by control LEDs [4].

Motor control: The stepper motor requires a combination of voltage states on its connection lines. The motor control has the function of generating and outputting these states. The stepper motor card installed in the WTD can operate motors up to 4 A in full, half, quarter and eighth step mode. The internal controller module *L6506* generates a code from the TTL frequency signal and the direction information, which it transfers to the motor. The direction information comes from the software and is placed on the bus of the electronics with a DAQ card. The frequency signal is generated by the frequency generator described in the next section. The generated signals are amplified by a power driver *L6203* and applied to the connection lines of the motor [4].

Frequency generator: The frequency generator specifies the operating frequency of the stepper motor. An alternating inverted TTL signal is required. In order to obtain this, two interconnected *NAND* components are used, which are operated as inverters (*NOT* components) [4].

Quadrature decoder: The quadrature decoder processes the incoming signals of the POSIMAG length measuring system and converts these into location information. The decoder forms a pulse signal which can be passed on to a counter. In the previous setup the total counter information was readout via a *Centronics* parallel interface with the PC. In the current setup, all counter informations are processed by the internal DAQ card [4].

Solenoid-controlled valve activation: The solenoid valve is opened by a 13 ms short pulse of 24 V. The pulse is generated by a monostable multivibrator, which is constructed with a *NE555* component. The circuit gets a start pulse, stays at a high level for a certain amount of time and then returns to the ground state. The short start signal at the trigger input of the chip is generated by a coupling-C combination and a transistor. A dynamic input signal is generated from a static input signal. An RC element defines the pulse duration at the respective inputs [4].

Optical sensor and preamplifier: The heart of the measuring system is an optical sensor from Hewlett Packard. It consists of a transmitter and a receiver for red light with a wavelength of 700 nm. An integrated optics focuses the light at a distance of 5 mm. The captured reflected light generates via a receiver photocell at the output of the sensor an electrical voltage between 0 V and 5 V. This voltage is applied to an operational amplifier component *LF351N*, which is connected as a non-inverting amplifier. The gain can be controlled by the ratio of resistances. The pre-amplified signal is sent to an analog input of the DAQ[5] card in the PC and analyzed by the software [4].

Interlock system: A limit switch system shuts off the frequency that powers the motor control, preventing further movement of the motor to prevent damage to the measuring device and the measuring object. There are two limit switches on the left side of the carriage and two more on the right side, which are each connected in series. If the carriage hits an obstacle, the switch is triggered upon contact. However, a single wire on the wire chamber is not enough to trigger the switch on contact. A careful handling of the wire chambers to be measured is therefore still required [4].

[5] **D**ata **A**c**Q**uisition

The previous setup included a PC with an AMD Athlon CPU with a clock speed of 1.466 GHz (single-core) and 256 MB RAM. The operating system (OS) was Windows XP (release 2001). The software for controlling the measuring equipment was LabVIEW[6] 7.0 (release 2003). In addition to the LabVIEW software, a National Instruments NI-PCI-MIO-16E-4[7] DAQ card with the *NI-DAQ* driver was used.

When working with this setup, especially the application to control the WTD, some difficulties arose. The *graphical user interface* (GUI) of this WTD application was divided into three programs. Each program had its own window. This was very cumbersome because for a simple command, for example to stop the stepper motor, you had to change quickly to another window.
Another big deficit of this application was the non-modular structure of the code. The libraries and subroutines were badly managed and were stored partly unrelated in a standard directory. In addition, many local variables were implemented which made it difficult to understand the code.
In the later course with this application it was noticed that most of the wires in a measurement were measured repeatedly. The consequences were a multiplication of the required measurement time and useless results. By mechanical action with the hand on the device it was possible to reduce this circumstance. It was assumed that the sensor did not properly detect the wires.
An important feature missing in this application was the ability to approach and remeasure specific wires. Only a recheck of a certain area by repositioning stoppers on the aluminum profile or repeating the entire measurement was possible.
A negative circumstance was that when a measurement was started, a time-out with a value of 30 minutes was activated in the background. When the limit switch of the probe carrier touched a stopper, the remaining time of the timeout had to be waited for the probe carrier to return to its original position. This had the disadvantage that either a long-term measurement could not be completed, or that had to be waited after a measurement.
All these negative aspects, especially the erroneous measurements, should be improved. In addition, the costs should be kept as low as possible. The methods will be described later in this bachelor thesis.

[6] **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench
[7] PCI interface, 500 kS/s sample rate (single-channel), 250 kS/s (multi-channel), 12-bit ADC resolution, 2 AO channels, 8 DIO channels, 2 counters

The previous setup was outdated and no longer adequate for today's standards. For the current setup a new PC with an Intel i5 processor with a basic clock frequency of 3.4 GHz (quad-core) and 16 GB of memory was purchased. The OS on this PC is Windows 10 and LabVIEW 2015 SP2 has also been installed. In addition a National Instruments NI PCIe-6320[8] DAQ card (see Fig. 3.5) with the latest *NI-DAQmx* drivers has been installed. NI-DAQmx incorporates an entirely new driver architecture and API[9], complete with new functions and development tools for controlling National Instruments DAQ devices. NI-DAQmx is entirely separate from the traditional NI-DAQ driver. The architectural changes and new features allow NI-DAQmx to deliver increased ease-of-use and improved performance over the traditional NI-DAQ driver [14].

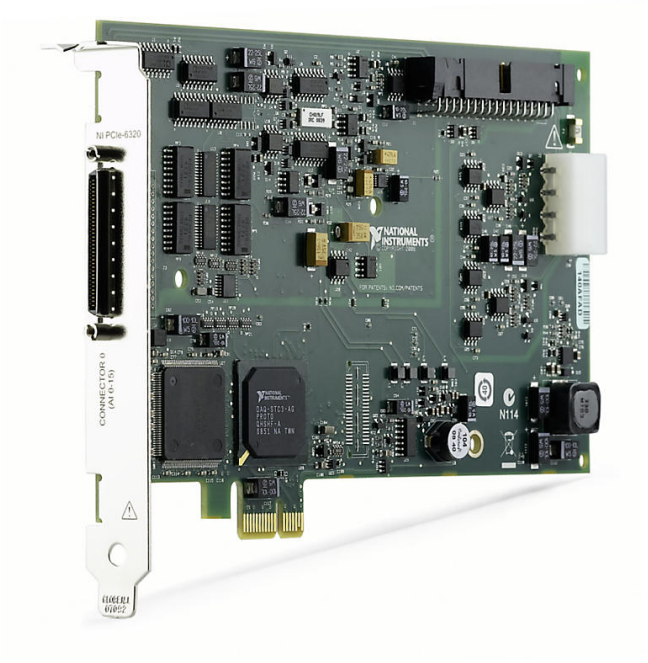

Figure 3.5: Image of the DAQ card NI PCIe-6320 [8].

[8] PCI-Express interface, 250 kS/s sample rate (single-/multi-channel), 16-bit ADC resolution, 8 differential or 16 single ended AO channels, 24 total DIO channels (8 + 16 PFI), 4 counters

[9] **A**pplication **P**rogramming **I**nterface

3.3 Working Principle

An optical sensor measures reflections on surfaces. It is capable of transmitting, receiving and converting this measurement into an electrical signal. The strength of the electrical signal provides informations about the reflection in the focal plane of the sensor (see Fig. 3.6).

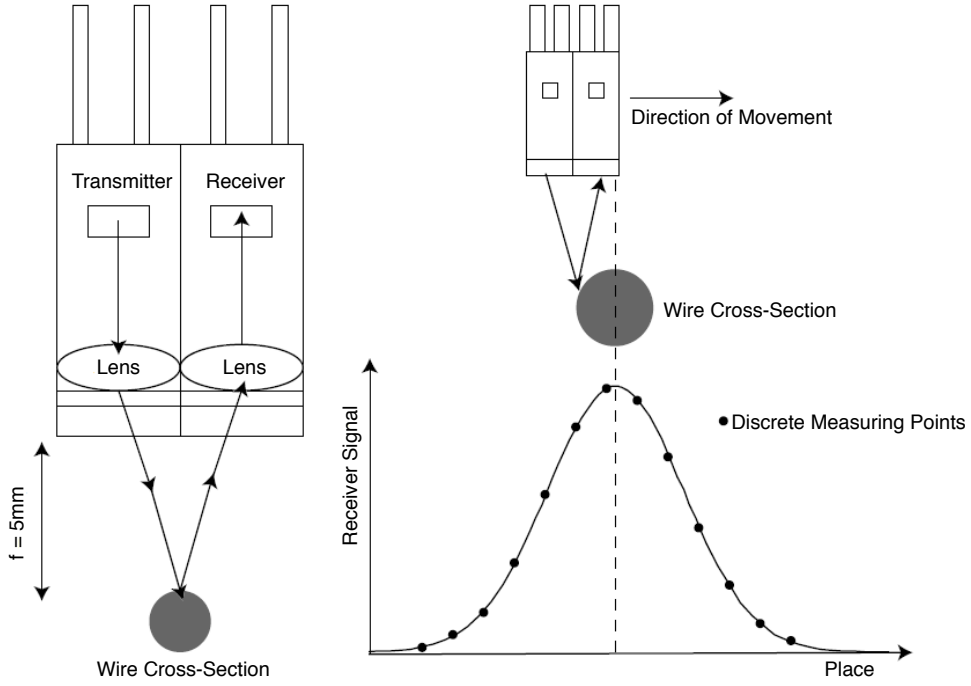

Figure 3.6: Functioning of the optical sensor. The sensor contains transmitter, receiver and a lens. If a reflected wire is brought into the focus of the optics, a growing voltage signal can be measured at the output of the sensor [4].

The principle of the measuring method is to bring the wire plane of the detector chamber into the focus of the sensor and to move across the chamber transversely to the wire direction (see Fig. 3.7).

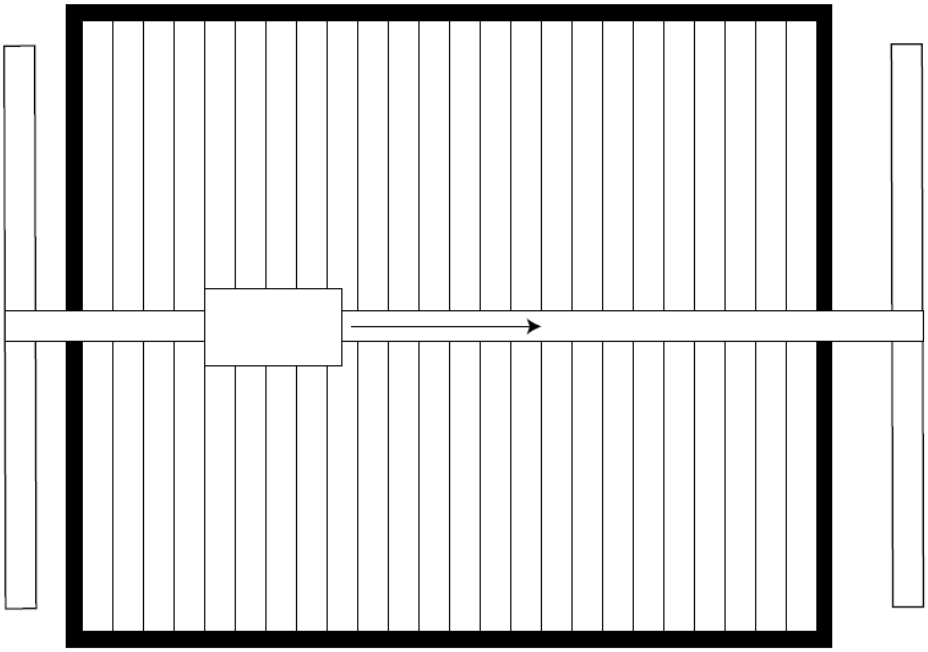

Figure 3.7: Direction of movement of the probe carrier in the top view. The probe carrier moves at right angles to the wire direction [4].

Since the presented method is based on reflection measurements, it must first be determined how strong the reflection signals of the wires are in comparison to the background noise signal. Due to possible oxidation of the CuBe wires and due to uneven gold plating of AuW wires, it turns out that the reflection signals can be very different from wire to wire. It is therefore necessary to compile a spectrum of the entire measurements with reflection signal amplitudes of each individual wire (see Fig. 3.8). You also want to get additional informations such as the number of wires found and their distances to each other. This is done with a so-called reference run. Subsection 4.3.1 describes this measuring process in more detail.

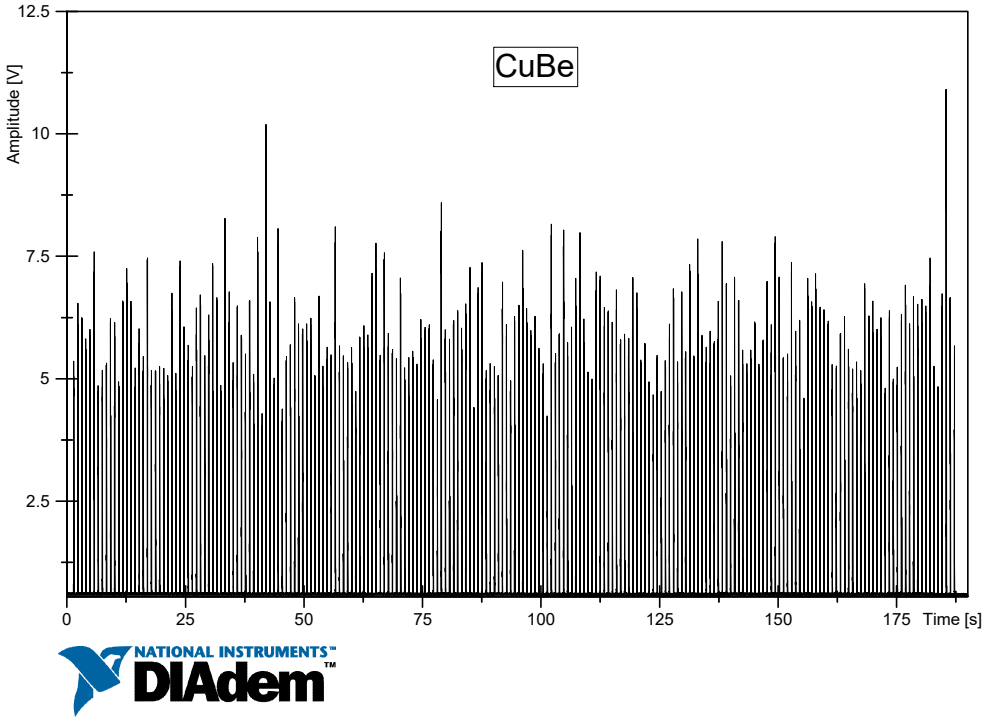

Figure 3.8: Screenshot of a reflection spectrum of CuBe wires. Clearly visible are the different reflection amplitudes of the individual wires.

The tension of the wires is measured separately with a so-called wire tension measurement. In this connection the photo sensor is positioned over a wire and then excited by a short air blast. The vibration of the wire at the output of the sensor can be detected as an oscillating voltage waveform (see Fig. 3.9).

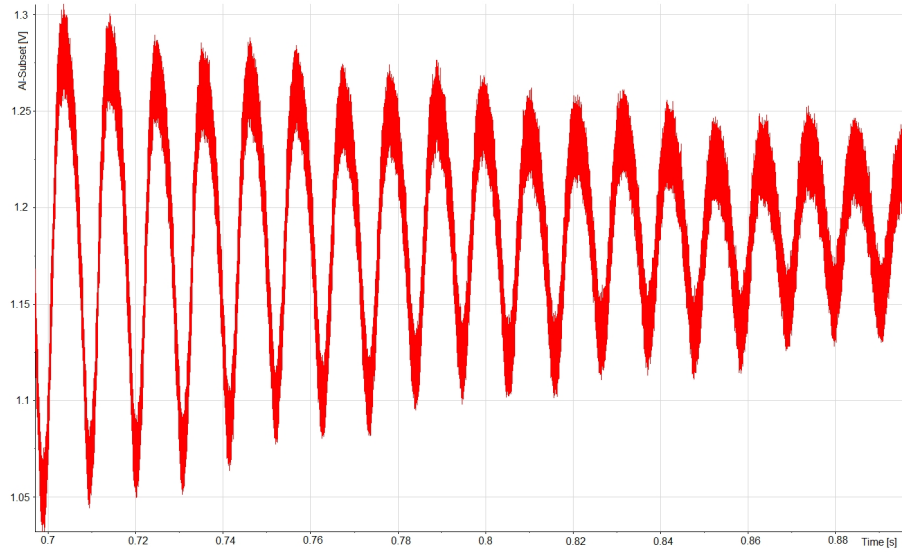

Figure 3.9: Screenshot of the oscillating waveform.

By means of a Fourier transformation, the frequency of the oscillation can be calculated from the amplitude profile. Taking into account the material parameters wire diameter, density and length, the mechanical tension of the wire can be determined. The following applies to the vibration modes of a string:

$$l = n \cdot \frac{\lambda}{2}, \tag{3.2}$$

with the length of the string l, the nth harmonic and the wavelength λ. Write for the frequency f and the propagation velocity v of a transverse wave along the string

$$f = \frac{v}{\lambda}, \tag{3.3}$$

so follows

$$f = \frac{n \cdot v}{2l}. \tag{3.4}$$

For v applies [6]:

$$v = \sqrt{\frac{F}{\rho \cdot A}}, \tag{3.5}$$

with the tension force F and the string cross-section A. If this is inserted in formula (3.4), this results to

$$f = \frac{n}{2l} \cdot \sqrt{\frac{F}{\rho \cdot A}} \tag{3.6}$$

or for the fundamental of the string ($n = 1$)

$$f_0 = \frac{1}{2l} \cdot \sqrt{\frac{F}{\rho \cdot A}} \tag{3.7}$$

and thus applies

$$F = 4f_0^2 \cdot l^2 \cdot \rho \cdot \pi \cdot r^2 \tag{3.8}$$

for the tension of the string of radius r. In order to determine the tension, the oscillation frequency of the wire and the known material parameters diameter, length and density are needed [4].

Another aspect is the power spectrum. In order to determine a frequency from an oscillation spectrum, a Fourier analysis is used. A Fourier transformation can convert a function from a time period $f(t)$ into a frequency space $g(\omega)$ [2]:

$$FT(f) = g(\omega) = \frac{1}{\sqrt{2}\pi} \int_{\infty}^{\infty} dt\ f(t) e^{i\omega t}. \tag{3.9}$$

In this case, t is the time variable and ω is the frequency variable. For the inverse transformation applies

$$FT(g) = f(t) = \frac{1}{\sqrt{2}\pi} \int_{\infty}^{\infty} d\omega\ g(\omega) e^{-i\omega t}. \tag{3.10}$$

Since the measurements do not provide a continuous function but a discrete, equidistant sequence of measurement points, one finds

$$g(\omega) = \sum_{i=1}^{N} \left(\frac{f_i}{N} cos\left(2\pi \cdot i \cdot \frac{\omega}{N} \right) \right) - i \sum_{i=1}^{N} \left(\frac{f_i}{N} sin\left(2\pi \cdot i \cdot \frac{\omega}{N} \right) \right). \tag{3.11}$$

To calculate this with the computer at runtime, there is a method based on a so-called *divide-and-conquer* algorithm. The idea behind this algorithm is to reduce the number of computation steps required by iteratively dividing the amount of data into roughly equal parts on which the problem is solved separately. From the solutions of the subproblems a solution of the total problem can be found. The measurements provide N data points for which N contributions are to be calculated (N sine and cosine calculations). The computation effort is therefore $\mathcal{O}(N^2)$. With the devide-and-conquer method, the effort can be reduced to $\mathcal{O}(N\ log_2\ N)$ calculation steps. If this method is applied to the Fourier transform of a data set of discrete measurement points, one can write:

$$g(\omega) = \sum_{j=0}^{N-1} e^{2i\pi j\omega/N} f_j \tag{3.12}$$

$$= \sum_{j=0}^{(N/2)-1} e^{2i\pi j\omega/(N/2)} f_{2j} + W(\omega) \sum_{j=0}^{(N/2)-1} e^{2i\pi j\omega/(N/2)} f_{2j+1} \tag{3.13}$$

$$= g(\omega)_g + W(\omega) \cdot g(\omega)_u, \tag{3.14}$$

in which g stands for the even and u for the odd t and $W(\omega) = e^{2i\pi\omega/N}$. Since only positive, non-imaginary parts should be considered, the power spectrum (PS) is used:

$$PS = \frac{FFT^*(signal) \cdot FFT(signal)}{N^2}. \tag{3.15}$$

In this case, FFT is the fast Fourier transform, FFT^* is the conjugated complex of the fast Fourier transform, and N is the number of measurement points in the examined data. The x-axis values of the peaks in the power spectrum correspond to the frequencies contained in the wire vibration [4].

If one wire is measured, the probe carrier moves to the next wire. It is also possible to selectively approach and measure individual wires. Subsection 4.3.2 describes this measuring process in more detail.

4 Software

4.1 LabVIEW

The control software for the WTD was written in the LabVIEW programming language. LabVIEW is the acronym for **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench and is a system-design platform and development environment for a visual programming language from National Instruments. It is differentiated by its graphical, general-purpose programming language known as *G* along with an associated integrated compiler, a linker and debugging tools. The advantage of a graphical programming language is that humans can capture a visual context better and faster than a cryptic program code.
Originally released for the Apple Macintosh in 1986, LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of operating systems, including Microsoft Windows, various versions of Unix, Linux, and Apple macOS. LabVIEW is different from most other general-purpose programming languages in two major ways. First, G programming is performed by wiring together graphical icons on a so-called block diagram, which is then compiled directly to machine code so the computer processors can execute it. While represented graphically instead of with text, G contains the same programming concepts found in most traditional languages, for example it includes all standard constructs, such as data types, loops, event handling, variables, recursion and object-oriented programming.
The second main differentiator is that G code developed with LabVIEW executes according to the rules of data flow instead of the more traditional procedural approach found in most text-based programming languages like C and C++. Dataflow languages promote data as the main concept behind any program. Its execution is data-driven, or data-dependent. The flow of data between nodes a program, not sequential lines of text, determines the execution order. Nodes in a LabVIEW program (in other words, functions, structures such as loops, subroutines and so on) have inputs, process data and produce outputs. Once all of a given node's inputs contain valid data, that node executes its logic, produces output data and passes that data to the next node in the dataflow path. A node that receives data from another node can execute only after the other node completes execution [7].

A LabVIEW program consists of a graphical user interface, the so-called *front panel* and the actual program code, the *block diagram* (see Fig. 4.1). The front panel contains all the control and display elements needed for the end user in virtual form, similar to a real measuring instrument. The internals of the measuring instrument can be found in the block diagram [18].

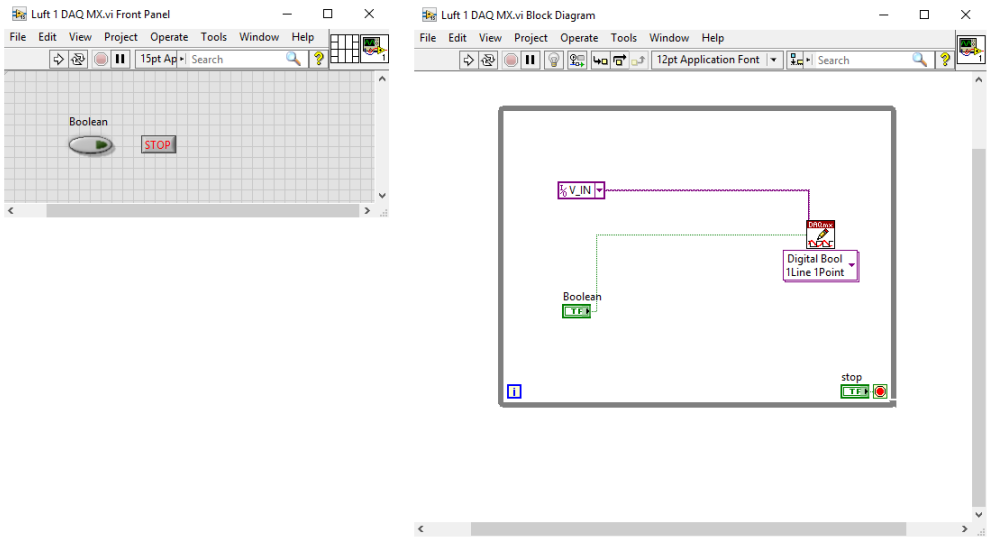

Figure 4.1: Screenshot of a simple VI for controlling the air supply. On the left the front panel and on the right the corresponding block diagram.

Because of these analogies, such a program is also called a *Virtual Instrument* (VI). VI files can be interleaved and invoked in subroutines. Such subroutines are called SubVIs. As already mentioned, a VI is written purely graphically instead of textual code. LabVIEW provides an extensive range of basic functions, data field operators, data filters, data analysis tools and much more. A program is created by selecting the desired functions and placing it on the workspace. To define a program sequence, the functions are connected by virtual wires.

Before it went to the actual encoding, the so-called *NI-MAX*[1] software had to be installed. NI-MAX provides access to the DAQ card and the individual ports of the electronics can be configured (see Fig. 4.2). Furthermore it can execute system diagnostics and run test panels.

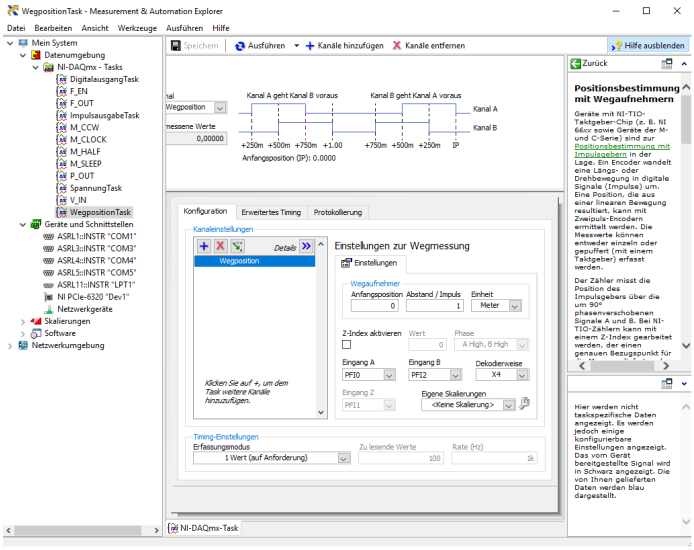

Figure 4.2: Screenshot of the NI-MAX GUI.

The programming environment used was the LabVIEW internal template *Queued Message Handler* (QMH). The QMH template facilitates multiple sections of code running in parallel and sending data between them. Each section of code represents a task, such as acquiring data, and is designed similarly to a state machine. Because of this design, it can divide each task into states. The QMH template is a version of a so-called *the Producer/Consumer* design pattern, where the user interface (producer) produces messages and the tasks (consumers) consume them (see Fig. 4.3). However, in the QMH template, it also can produce messages from a consumer loop. The default QMH template includes one producer loop and one consumer loop [10]. Additional consumer loops were added in the final code. The advantage of this architecture is that it is modularly expandable. For future hardware and software updates any consumer loops can be edited or new ones can be added.

[1] **N**ational **I**nstruments **M**easurement & **A**utomation E**X**plorer

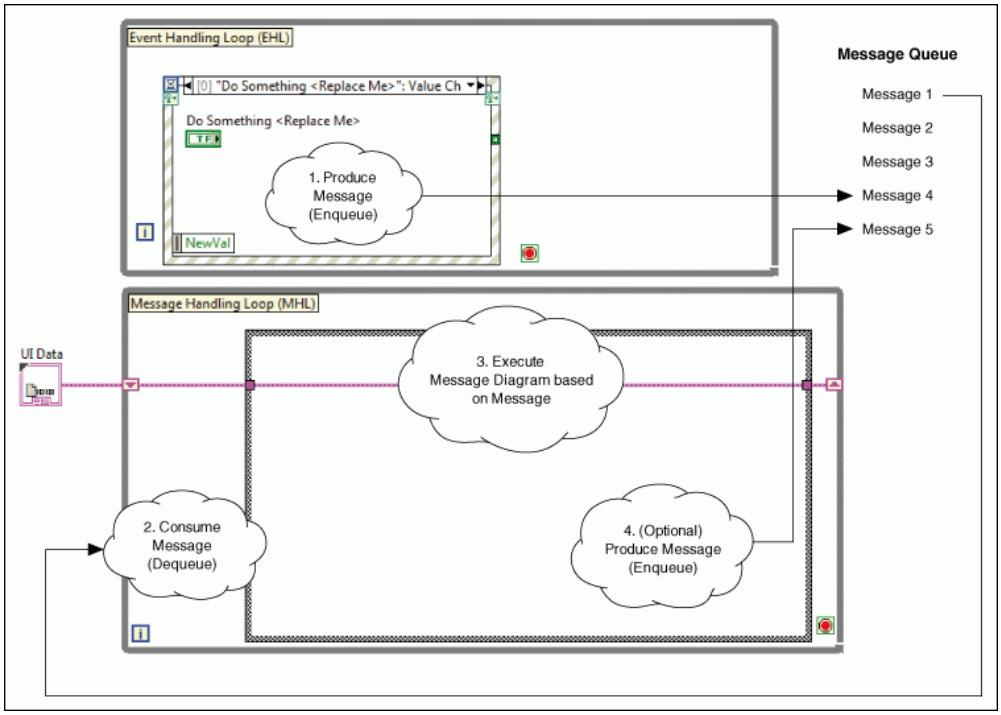

Figure 4.3: Screenshot of the LabVIEW Queued Message Handler Template. This template repeatedly executes the following steps [11]:

1. A user interacts with the front panel, causing the event structure in the "Event Handling Loop" (EHL) to produce a message. LabVIEW stores the message in a queue.

2. The "Message Handling Loop" (MHL) reads a message from the message queue, removing the message.

3. The message is a string that matches one of the subdiagrams of the case structure in the MHL. Therefore, reading the message causes the corresponding subdiagram of the case structure to execute. This subdiagram is called a message diagram because it corresponds to a message.

4. Optionally, the message diagram produces another message, storing it in the message queue.

The GUI of the previous WTD application was divided into three VIs. Each VI had its own window. This unnecessarily complicated the operation. This factor is no longer included in the current code. All important controls are combined in one VI (see Fig. 4.4).

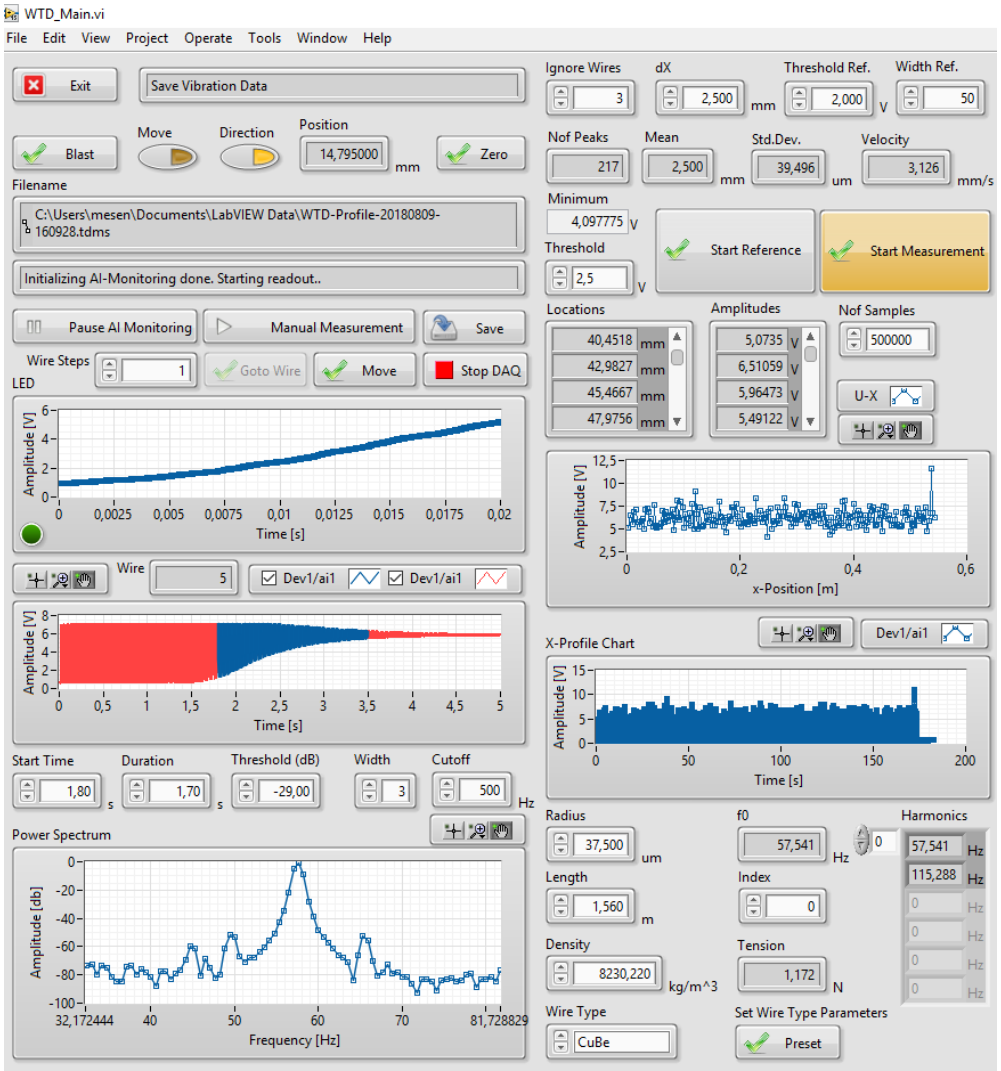

Figure 4.4: Screenshot of the GUI. In the appendix the functions are described in more detail.

A big deficit of the previous WTD application was the non-modular structure of the code. The libraries and subroutines were badly managed and were stored partly unrelated in a standard directory. In addition, many local variables have been implemented in the code. This type of programming brings many disadvantages. On the one hand the clarity suffers and on the other hand local variables require memory space per variable to be read. If the variable is a complex cluster or even an array, the space requirement is correspondingly larger. In addition to the memory requirements, there is of course also the processor time that is consumed. All these negative factors have been corrected in the current code. The main focus was a modular design, which is also prepared for future updates and upgrades. Subroutines are much easier to edit or completely replace- and integrable. Most of the local variables were omitted. Fig. 4.6 gives a visual overview of the complete block diagram.
All necessary libraries and subroutines are arranged via a so-called "LabVIEW project" and can be easily selected (see Fig. 4.5).

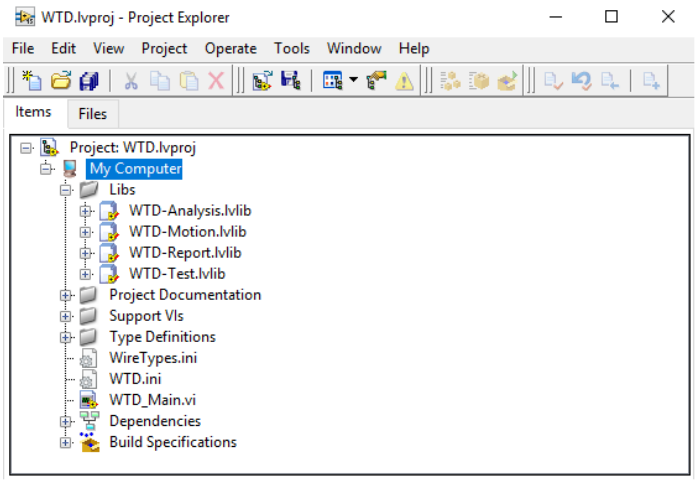

Figure 4.5: Screenshot of the LabVIEW Project Explorer.

Files are stored in the so-called TDMS[2] format. The binary TDMS file format is an easily exchangeable, inherently structured, high-speed-streaming-capable file format that becomes quickly searchable without the need for complicated and expensive database design, architecture, or maintenance [12].
To analyze the measured values, the software *DIAdem* is used. DIAdem is a single, unified software environment, which is optimized for TDMS files. It includes tools to view and investigate data sets [13]. TDMS files can also be opened with spreadsheet programs such as Microsoft Excel. You can also copy the measured values from the TDMS file and save them in ASCII[3] format.

[2] **T**echnical **D**ata **M**anagement **S**treaming
[3] **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange

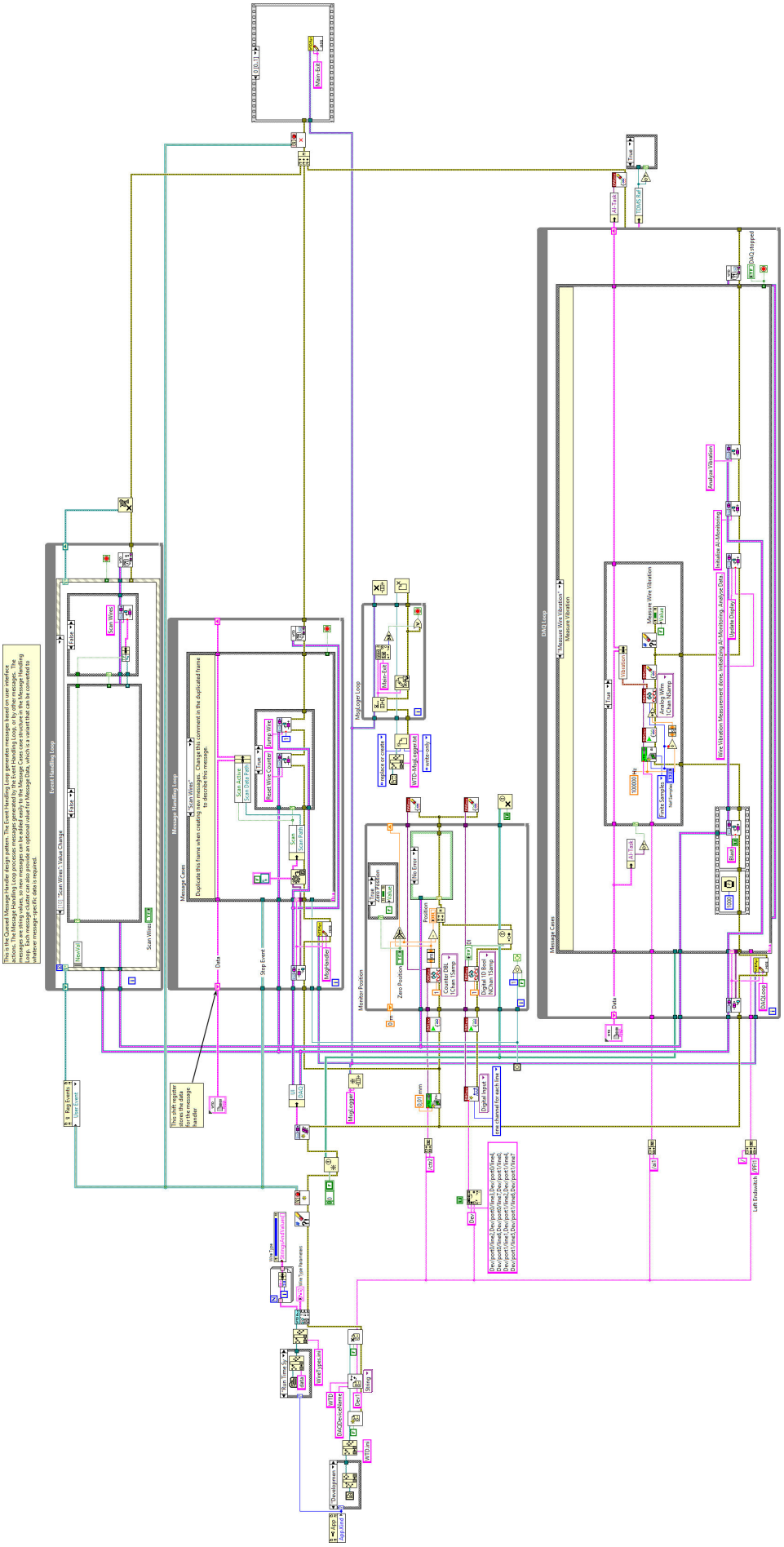

Figure 4.6: Screenshot of the complete block diagram.

4.2 Technical Issues and General Improvements

While measuring it was noticed that the length measuring system POSIMAG was faulty because it had a kind of slip. If the probe carrier was moved from the zero position back and forth, the system indicated a deviation. This deviation depended on the measured length and the speed of the stepper motor. Both, increasing the speed, as well as a longer measuring section resulted in a higher deviation. Deviations of up to 7.5 mm could be observed, which made an exact position determination of the wires almost impossible. Most of the wires in a measurement were measured repeatedly (up to ten times) because the sensor did not properly detect the wires. The consequences were a multiplication of the required measurement time and useless measurement results. By mechanical action with the hand on the device it was possible to reduce this circumstance. The signal of the POSIMAG was monitored via an oscilloscope, in this case impure TTL signals could be observed (see Fig. 4.7), which could be responsible for the slip.

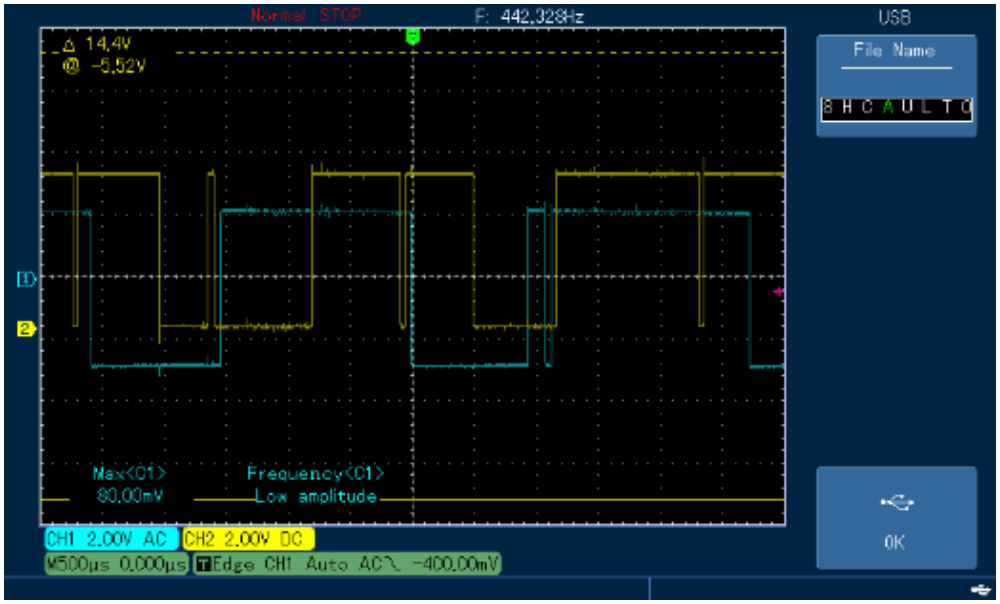

Figure 4.7: Screenshot of the TTL signal taken from an oscilloscope. Clearly recognizable is the impure sinusoidal signal compared to figure 3.2.

Since the POSIMAG was unusable for exact position determination, there were only two options: either install a new length measuring system, which would be associated with high costs or use an alternative method. It was decided for the alternative method. The stepper motor runs at a constant speed and the DAQ card works with a sampling rate of 1000 Hz. Because of this constant sampling rate, time stamps can

be assigned to the measurement results. If the speed was known, it could be used to calculate the positions. One can calculate the speed, if one knows the distance between two measured time stamps. A *dX* parameter was implemented specifically for the reference run. This parameter specifies the distance between two adjacent wires. As already mentioned, the reference run counts the found wires. The algorithm calculates the distance between the first and the last considered wire. With the timestamps and the distance between two wires, the speed can now be determined. With this information, the code can now calculate the wire positions. It should be kept in mind that this method works with a given reference length and is therefore only as accurate as the reference length itself. Should the winding machine accidentally wind at 5 mm intervals, the code will not notice and you can not do so from the results. Therefore, this reference length should be cross checked. Although the POSIMAG is no longer used to calculate the wire positions, it still exists in the code. For a simple cross-check, you can use it by measuring the distance between the first and last considered wire. A small example: If one had a wire winding with 2.4 mm instead of 2.5 mm intervals for a number of 101 wires, the deviation would be 10 mm to 250 mm measuring distance. The POSIMAG would expect values in the range of 250 mm ±2 mm. In fact, you should get values in the range of 240 mm ±2 mm. This discrepancy should be noticed. Alternatively, you can measure the distance manually, but this method should be more accurate than the POSIMAG. If one has the exact distance between the first and the last wire, one can divide this value by the number of the wire distances and use it for the dX parameter.
Under the premise that the winding machine works reliably, this implemented dX parameter method is straightforward and accurate. Neither a manual measurement of the reference length nor a manual counting of the wires is necessary. Rough deviations can be detected with a POSIMAG cross check. For even more accurate results you need an accurate measurement of the reference length, there is no alternative.
In addition to the function of POSIMAG a half-step mode of the stepper motor was implemented in the previous code to let the sensor detect the wires properly. This resulted in greater sound volume and also consumed more time because the probe carrier was slowed down when approaching a wire. In the current code the half-step mode was omitted. In combination with the new measuring method without the faulty POSIMAG, the measuring time could be drastically shortened and, as a side effect, the sound volume could be reduced.

There was another problem during the measurements. It was noticed that at each wire tension measurement a noise frequency of 50 Hz appeared in the power spectrum (see Fig. 4.8). It was obvious that this had to do with the frequency of the electricity grid. By switching off multiple power strips around the laboratory bench, it was possible to observe the disappearance of this noise frequency. It was important to eliminate this noise source, because the searched frequencies can be in the scope of this 50 Hz peak.

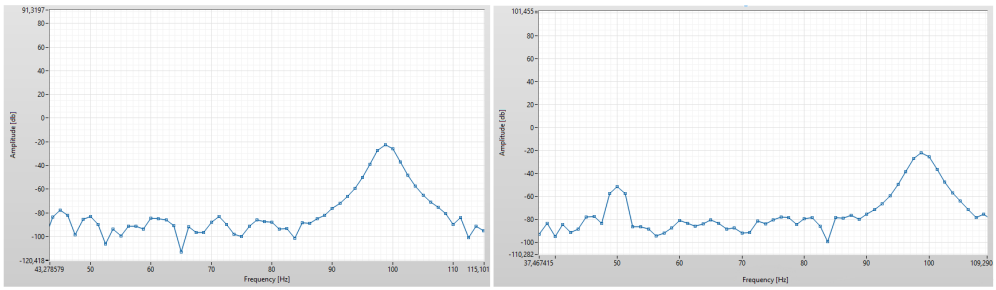

Figure 4.8: Shown here is the effect of the power strips around the laboratory bench. Left picture powered off, right picture powered on power strips. The peak at 50 Hz is clearly visible (taken from GUI).

The possibility to approach and measure a specific wire was implemented with a "Move to" function. The 30 minute time-out included in the previous code no longer exists. The probe carrier moves back to its original position after contact with the stopper. The reference run and the wire tension measurement were connected in the previous code. This means that with each measurement, both routines were processed, which consumed additional time, even if only one of the measurements was needed. In the current code both measuring methods are separated. The operator can now choose for himself which measurement he wants to perform.

4.3 Algorithm and Measuring Procedure

The measurement is divided into two parts: the preparation in the form of a reference run and a subsequent wire tension measurement. Both measuring methods are described in more detail in the following subsections.

4.3.1 Reference Run

The reference run is intended to measure the reflections, the locations of the wires, the distances between them and to count the number of wires found. This is important to see in advance if the positions of the wires are correct or if there are other irregularities such as a missing or hanging wire. In addition, the program provides information regarding the mean value of the wire distances, the standard deviation of the wire distances, the amplitude values, the speed of the stepper motor and various plots.

After starting the "WTD_Main.VI" the GUI of the WTD application appears. The "AI-Monitoring" (analog input) of the photo sensor, all DAQmx tasks are initialized and the readout begins. The information from the photo sensor is forwarded to the LED chart (see Fig. 4.9). It should be mentioned that the DAQ card works with a sampling rate of 1000 Hz during the reference run.

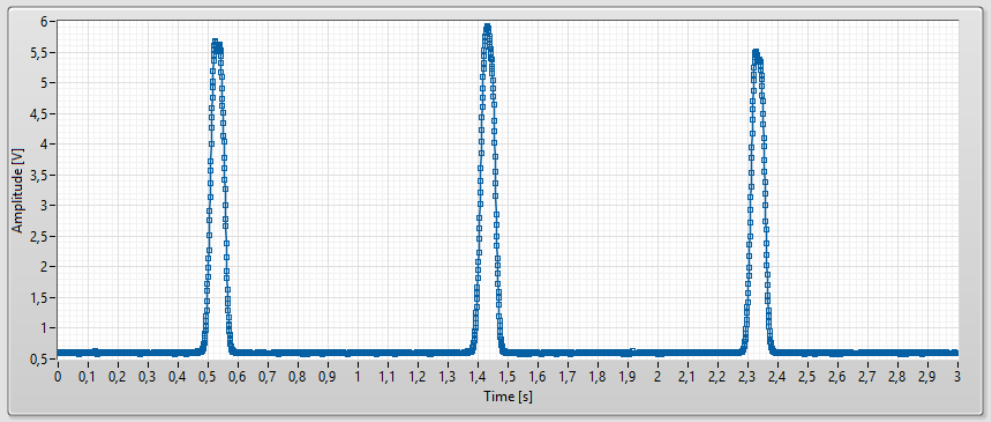

Figure 4.9: Screenshot of the LED chart (taken from GUI). The captured reflection of the wire generates an electrical voltage via the receiver photocell at the output of the sensor. This signal is then forwarded to the analog input of the DAQ card and analyzed by the software.

Before the actual measurement, the height of the photo sensor should be checked and adjusted if necessary. For CuBe wires a reflection value of about 5 V is recommended and for AuW wires about 2 V.

Depending on which wire type and chamber or frame is used, you can select predefined presets under the drop down menu "Wire Type". A so-called *INI* file, which is stored in the LabVIEW project is accessed in this process. INI is a file extension for an initialization file format. INI files are plain text and are used to set parameters for some programs. By pressing the button "Set Wire Type Parameters", these presets are accepted. These parameters can also be entered manually.
The reference run can be started by pressing the "Start Reference" button. The program then wants to create a TDMS file, in which all required measurement results of the reference run are stored. After creating the file, the case structure "Reference Start" is initiated. DAQmx tasks of the stepper motor and the POSIMAG are accessed and the probe carrier is moved to the left stopper. The current position of the measuring head is determined by continuously reading the POSIMAG with the DAQ card's internal counter and is displayed in the GUI. After contact with the limit switch the position counter is automatically set to the value 0.
The command for the stepper motor to move to the right until the stopper is reached is given. The continuous loops "Readout" of the photo sensor and "Update Display" for the LED chart are running in the background. The program now takes the data from the photo sensor and stores it as raw data in the previously created file.
Once contact to the right stopper is made, the retraction of the probe carrier is activated. At the same time the case structure "Find Wires" will then be executed.
Heart of this case structure is the so-called SubVI *Peakfinder*, which includes the LabVIEW function *Peak Detector.VI* (owning palette: *Signal Operation VIs*). The parameters "Ignore Wires" and "dX" flow into the process. The "Ignore Wires" function ignores a certain number of wires in the reference run. Since the wires are not optimally tensioned at the beginning, as well as at the end of a winding, this function can be used if necessary. The ignored wires are not taken into account for the later calculation of the standard deviation. The dX parameter was discussed in the previous section 4.2.
Peak Detector.VI is based on an algorithm that fits a quadratic polynomial to sequential groups of data points. The number of data points used in the fit is specified by the parameter "Width Ref.[4]", which depends on the wire diameter and thus is a known constant. For each peak or valley, the quadratic fit is tested against the parameter "Threshold Ref.". If the threshold is specified too low, low-level peaks will be detected from the background noise level, if the threshold is specified too high, not all wires can be found [9].
As one can see in Fig. 4.10, some peaks have a dent. In this case, it may happen that the Peak Detector.VI recognizes this as two peaks. To prevent this, a value must be set for the width that is so large that the range taken into account by the Peak Detector.VI includes both maxima of the double peak. It is recommended to analyze the amplitudes after the reference run and, if necessary, to adjust the values for "Width Ref." and to repeat the reference run.

[4] Ref. is the abbreviation for reference run. This note in the GUI is intended to prevent confusion with the wire tension parameters.

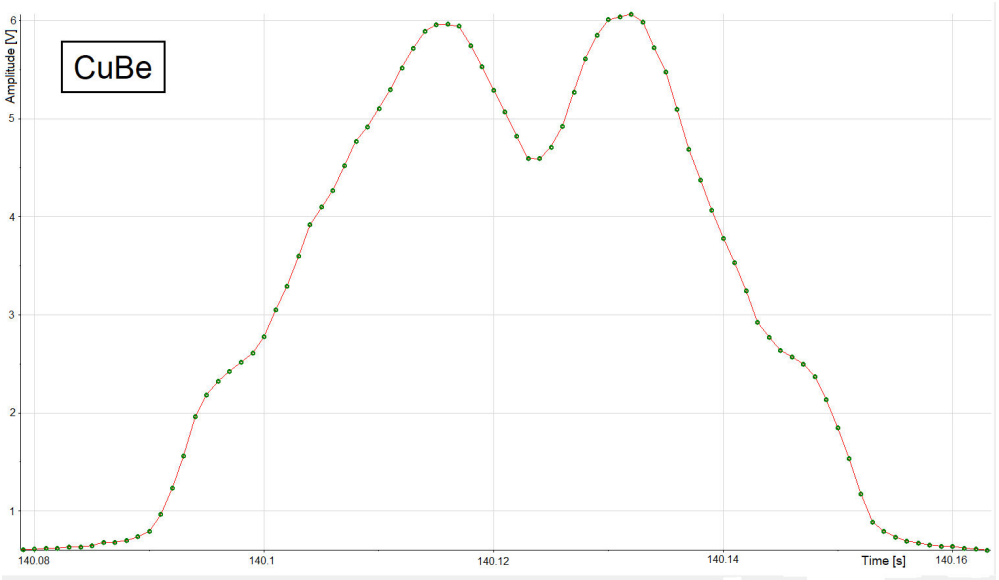

Figure 4.10: This image comes from the reference run of CuBe wires. Enlarged, it shows the double peak, which can be recognized by the Peakfinder as two wires. There are about fifteen measuring points between the two peaks. Therefore, a width of 25 is recommended for this type of wire.

The Peakfinder determines the peaks from the recorded raw data of the photo sensor and the number is then displayed in the GUI under the "Nof Peaks" indicator. In addition, "Mean Value of Wire Gaps", "Standard Deviation of Wire Gaps", "Minimum Amplitude" and "Velocity" are output and shown in the GUI. Based on the found minimum amplitude, the Peakfinder proposes a threshold for approaching the wires in the subsequent wire tension measurement (this threshold value is explained in more detail in subsection 4.3.2). Furthermore, the indices "Amplitudes" and "Locations" and the corresponding plots are displayed graphically in the GUI for a quick check of the measurement (see Fig. 4.11 and 4.12). All relevant data are stored in the file and then finalized after completion of the measurement (see Fig. 4.13). Afterwards the WTD application can be left with the "Exit" button. For a better illustration, flowchart 4.14 is used.

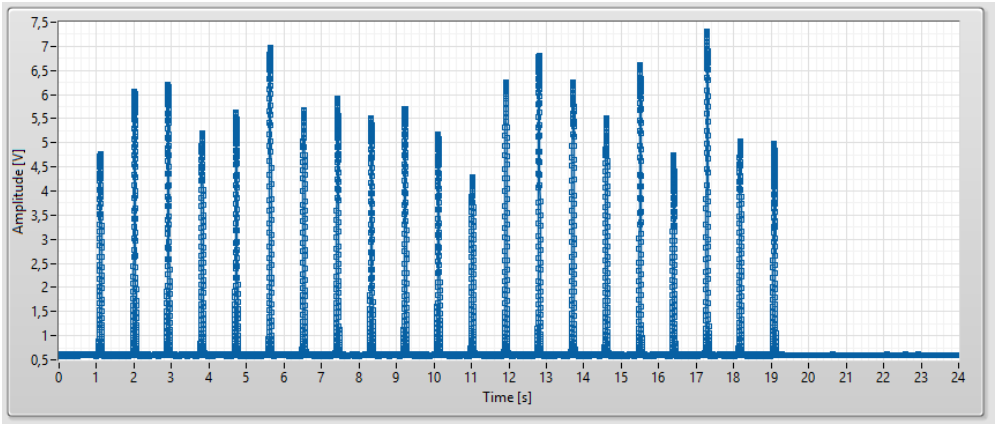

Figure 4.11: Screenshot of the "X-Profile" chart, which shows the raw data processed by the Peakfinder (taken from the GUI). The voltage [V] is plotted against time [s].

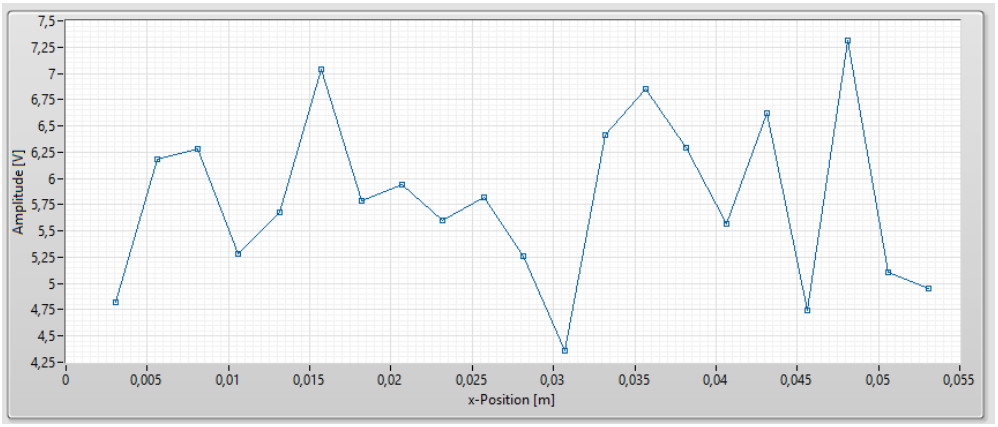

Figure 4.12: Screenshot of the "U-X" chart after the Peakfinder has performed its calculations (taken from the GUI). Taking into account the determined speed, this chart shows the position [m] of the individual wires plotted against voltage [V].

LED	Locations	Amplitudes	Gaps
0,582093075	0,001695828	1,438660212	0,002571791
0,581764219	0,004267619	1,593803886	0,002514272
0,576502512	0,006781891	1,569793315	0,002496192
0,579133365	0,009278083	1,253929641	0,00250371
0,578475652	0,011781793	1,481580144	0,002520907
0,579133365	0,0143027	1,423963556	0,002495262
0,597549339	0,016797961	1,486303514	0,002484912
0,579791079	0,019282874	1,261766347	0,002507651
0,586368212	0,021790525	1,593742851	0,002499394
0,579791079	0,024289919	1,653530822	0,002514583
0,583737359	0,026804502	1,454819672	0,002473409
0,579791079	0,029277911	1,306456024	0,00250243
0,580448792	0,031780341	1,442207342	0,002496334
0,585710499	0,034276675	1,248195769	0,002460123
0,575515942	0,036736798	1,588221195	0,002486776
0,584395072	0,039223574	1,613526245	0,002505519
0,578804509	0,041729093	1,364222635	0,002509004
0,581764219	0,044238097	1,354562889	0,002415807
0,578146795	0,046653905	1,784877933	0,00253905
0,582093075	0,049192955	1,418652753	0,002502873
0,580119935	0,051695828	1,164262746	0
0,578475652			
0,593603059			
0,576831368			
0,581764219			
0,575515942			
0,579133365			
0,579462222			
0,578475652			
0,588341352			
0,579462222			
0,586368212			
0,580448792			
0,583737359			
0,579791079			

Figure 4.13: Screenshot of all relevant tables, which are stored in the TDMS file. In the "LED" column, the raw data of the photosensor are stored (in this case there were 36000 entries). From these raw data, the columns "Locations", "Amplitudes" and "Gaps" are calculated by the Peakfinder.

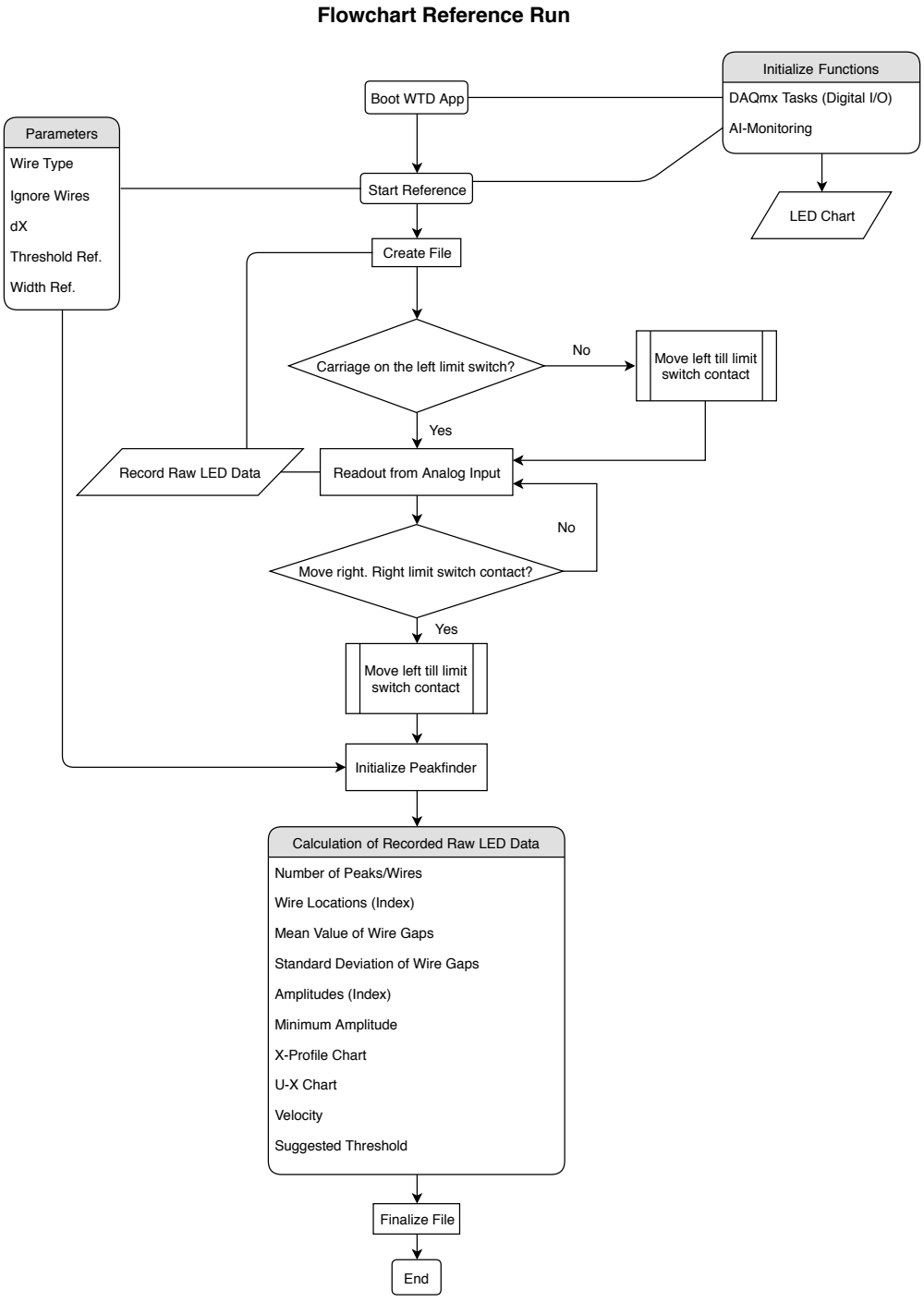

Figure 4.14: Simple flowchart of the reference run.

4.3.2 Wire Tension Measurement

As with the reference run, the WTD application must first be booted. Before the wire tension measurement can be started, the probe carrier must be moved manually to the left stopper if necessary. Material parameters of the wire properties ("Diameter", "Length" and "Density") for the later calculation of the wire tension, parameters for the Fourier transform ("Start Time" and "Duration") and parameters for the Peak Detector.VI ("Threshold", "Width" and "Cutoff") are required. This can be done manually or using predefined values via the "Wire Type" tab. In addition, a threshold for approaching the wires is needed. This threshold value can be used from the reference run or the recommended values are 2.5 V for CuBe wires and 0.85 V for AuW wires. The complete measuring method will be described in more detail in this subsection.
The wire tension measurement is activated by pressing the "Start Measurement" button and the program then wants to create a TDMS file in which all acquired measurement results of the wire tension measurement are stored. Then the case structure *Jump Wire* is initialized. DAQmx tasks of the stepper motor and the POSIMAG are accessed and the probe carrier is moved to the right. The current position of the measuring head is determined by continuously reading the POSIMAG with the DAQ card's internal counter and is displayed in the GUI. The POSIMAG function is still included in the final code because the data is sufficient for a cross-check. The later created column "Position" (see Fig. 4.18) represents the values of the POSIMAG. If a wire is skipped during the measurement, this column can be used to determine which wire was skipped. In the previous code, a "Peak-File" was created from the reference run, in which all peak positions of the wires determined by POSIMAG, were stored. The software accessed this file in the wire tension measurement to approach the wires by comparing the current reading of the sensor with this data. This method was rejected and replaced by a new one.
To approach the wires the SubVI *Check for Leading Edge* was written and implemented in the case structure Jump Wire. As the probe carrier moves, the signal from the photo sensor is written into a buffer with 250 kHz. Once the buffer contains 2500 samples these are forwarded to Check for Leading Edge, which adjusts a regression line. If the initial value is below the threshold and the final value above, then this is recognized as the edge of a wire signal and the probe carrier stops. With this method, the edges of the wire signal are reliably detected. This process runs in real time.
The photo sensor is thereby positioned over a wire and the software executes the case structure *Measure Wire Vibration*. The DAQmx task for the pressurized gas supply is accessed and the wire is excited by a short air blast. The resulting vibration of the wire is read out at the output of the photo sensor as an oscillating voltage waveform, which is forwarded to the subset chart (see Fig. 4.15). The DAQ card works in this method with a sample rate of 100 kHz. The already mentioned parameters "Start Time" and "Duration" flow into this process and determine the range in which the Fourier transform is applied.

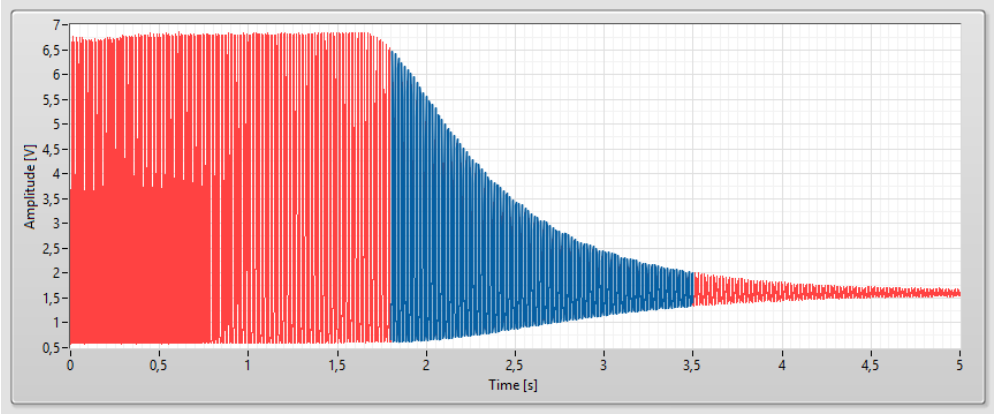

Figure 4.15: Screenshot of the oscillating voltage waveform in the subset chart (taken from GUI). The blue subset is the area in which the Fourier transform is applied.

A subset (marked in blue) is formed with these parameters which is passed to the case structure *Analyze Vibration* where the SubVI *FFT* is located. This SubVI calculates the power spectrum with this subset taking into account the formula (3.15) and this is shown in the power spectrum chart (see Fig. 4.16).

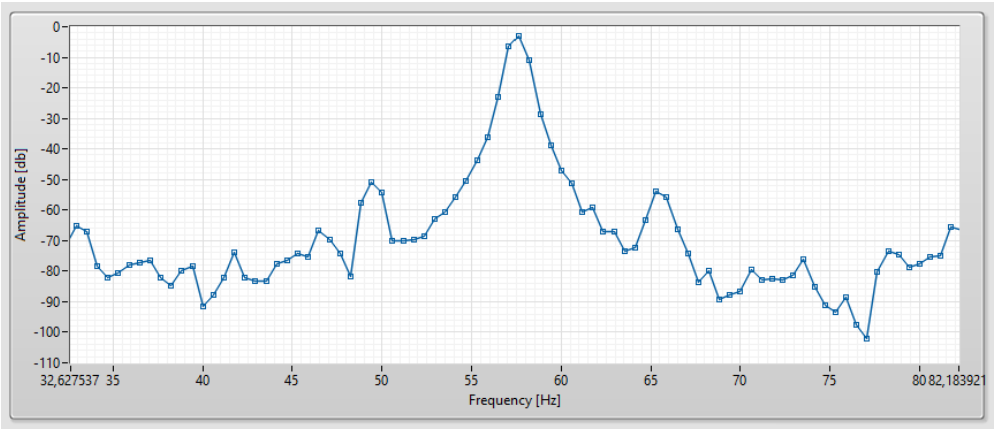

Figure 4.16: Screenshot of the power spectrum chart of a measured CuBe wire (taken from GUI). The dominant peak is located at 57.54 Hz, which would correspond to a wire tension of 1.17 N.

If a valid time axis information is added to the power spectrum, the frequencies of the examined wire vibration can be read in Hz. For the wire tension calculation, the fundamental frequency f_0 of the wire vibration is required (see section 3.3). This is the dominant peak in the power spectrum.
In the software, this peak is determined by the Peak Detector.VI, which fits local quadratic functions to the data at runtime. The parameters "Threshold", "Width" and "Cutoff" flow into this process and are described in more detail in subsection 4.3.1. The parameter "Cutoff" defines the limit of frequencies, which are considered by the Peak Detector.VI for the calculation. The value of the fundamental is displayed in the GUI.
The determined fundamental and the wire parameters "Radius", "Length" and "Density" are then forwarded to the *Wire Tension.VI* (see Fig. 4.17), which includes the formula (3.8).

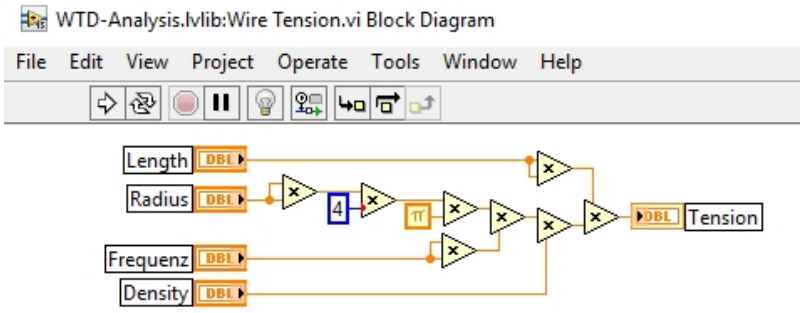

Figure 4.17: Screenshot of the Wire Tension.VI. This block diagram corresponds to the formula (3.8).

This VI finally calculates the wire tension, displays the value in the GUI and writes it to the file along with the other relevant informations like the corresponding wire number, all relevant parameters, the fundamental vibrations, other found harmonics, the amplitude values, the subsets and the power spectra (see Fig. 4.18). Then the case structure Analyze Vibration is left and the process Jump Wire is repeated.
When the right stopper is reached, the measurement stops and the probe carrier has to be manually returned to its original position. Afterwards the WTD application can be left with the "Exit" button. For a better illustration, flowchart 4.19 is used.

Group	Channels	Description	Cutoff_Hz	Density_kg_m_3	f_max_Hz	f_User_Hz	Length_m	Peak_Index	Position_m	Radius_m	Tension_N	Threshold	Width
Wire_0	4		500	19289,579	94,98997886	94,98997886	1,56	0	0,001725	0,00001	0,532276622	-35	3
Wire_1	4		500	19289,579	94,02625117	94,02625117	1,56	0	0,004265	0,00001	0,521530909	-35	3
Wire_2	4		500	19289,579	93,75651533	93,75651533	1,56	0	0,006775	0,00001	0,518542939	-35	3
Wire_3	4		500	19289,579	94,90107218	94,90107218	1,56	0	0,009275	0,00001	0,531280711	-35	3
Wire_4	4		500	19289,579	94,74117324	94,74117324	1,56	0	0,01178	0,00001	0,529491908	-35	3
Wire_5	4		500	19289,579	95,13831563	95,13831563	1,56	0	0,014275	0,00001	0,533940331	-35	3
Wire_6	4		500	19289,579	94,52800744	94,52800744	1,56	0	0,016775	0,00001	0,527111895	-35	3
Wire_7	4		500	19289,579	94,20057573	94,20057573	1,56	0	0,019285	0,00001	0,523466537	-35	3
Wire_8	4		500	19289,579	95,65808492	95,65808492	1,56	0	0,021745	0,00001	0,539790422	-35	3
Wire_9	4		500	19289,579	93,59045089	93,59045089	1,56	0	0,024255	0,00001	0,516707647	-35	3
Wire_10	4		500	19289,579	93,41070526	93,41070526	1,56	0	0,02676	0,00001	0,514724822	-35	3
Wire_11	4		500	19289,579	94,54646765	94,54646765	1,56	0	0,029265	0,00001	0,527317793	-35	3
Wire_12	4		500	19289,579	93,49979507	93,49979507	1,56	0	0,03176	0,00001	0,515707121	-35	3
Wire_13	4		500	19289,579	93,43781301	93,43781301	1,56	0	0,03428	0,00001	0,515023611	-35	3
Wire_14	4		500	19289,579	94,04512592	94,04512592	1,56	0	0,036725	0,00001	0,521740313	-35	3
Wire_15	4		500	19289,579	92,69269249	92,69269249	1,56	0	0,03923	0,00001	0,506842244	-35	3
Wire_16	4		500	19289,579	92,56041885	92,56041885	1,56	0	0,04176	0,00001	0,505396735	-35	3
Wire_17	4		500	19289,579	93,34161788	93,34161788	1,56	0	0,044275	0,00001	0,513963713	-35	3
Wire_18	4		500	19289,579	93,50844464	93,50844464	1,56	0	0,04669	0,00001	0,51580254	-35	3
Wire_19	4		500	19289,579	92,17970598	92,17970598	1,56	0	0,04925	0,00001	0,501247763	-35	3
Wire_20	4		500	19289,579	57,83362219	57,83362219	1,56	0	0,051775	0,00001	0,197307047	-35	3

Figure 4.18: Screenshot of all relevant tables from the wire tension measurement. All important information such as the wire tensions and the corresponding wire numbers as well as all set parameters are stored here.

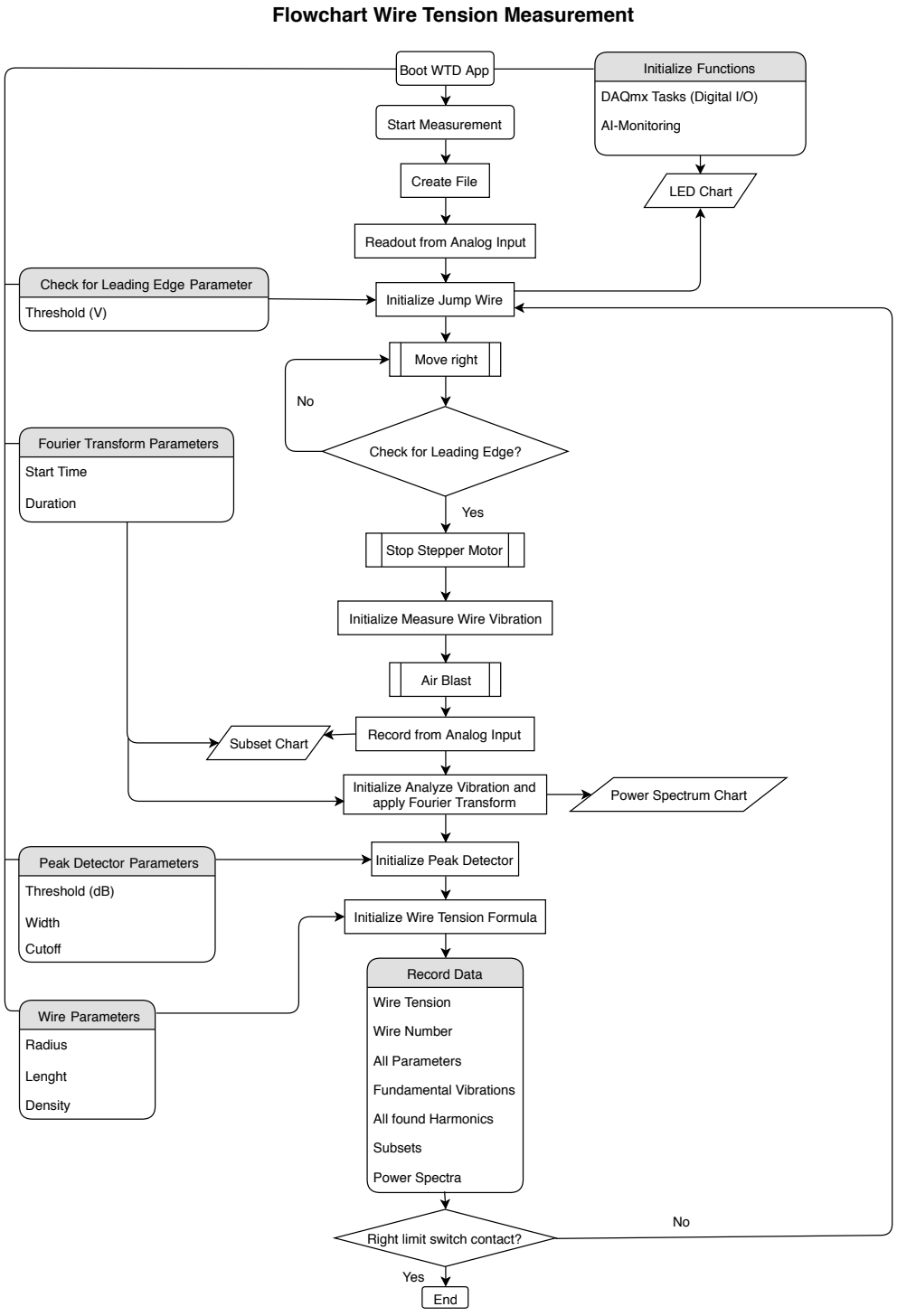

Figure 4.19: Simple flowchart of the wire tension measurement.

5 Measurements and Results

Measurements were performed on so-called frame windings. Two CuBe windings and one AuW winding were measured. One of the CuBe windings was not optimally wound due to incorrect handling of the winding machine. The stoppers on the aluminium profile were used to define a measuring section with 21 wires and consequently 20 wire distances. In order to get sufficient statistics, the wire distances and wire tensions were measured 100 times each. The non-optimal CuBe winding was measured only once. The analysis of the measurement results was done with the *ROOT Framework*. At this point, attention should be drawn to the small error bars in Figures 5.2, 5.4, 5.7 and 5.10. They are each based on 100 separate measurements from which the mean value and the associated standard deviation were calculated. This refers to both the measurement of the individual wire tensions, as well as the wire distances. The variations in the tensions and distances of the various wires around the indicated average value are based on actual differences in the wires. These variations have no significance for the measurement of a single wire.

5.1 CuBe-Wire

For the CuBe measurement, a frame winding was wound with a target value of 1.2 N for the wire tension and 2.5 mm for the wire distances.

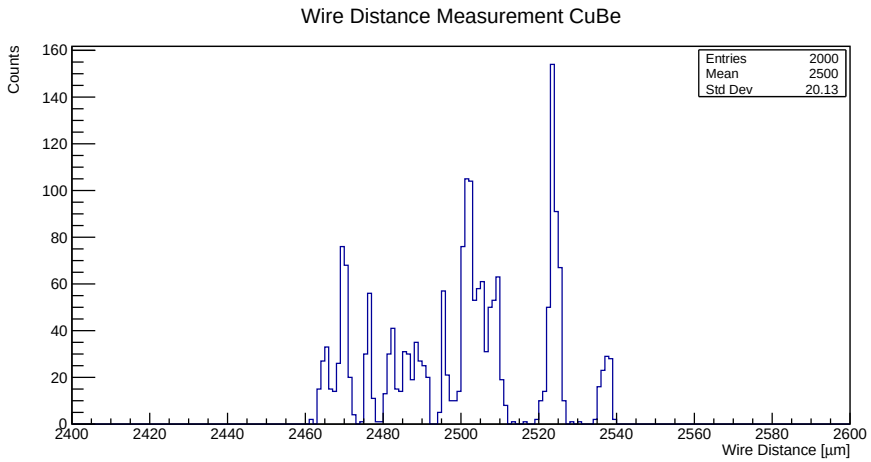

Figure 5.1: 100 distance measurements of two consecutive wires for all 20 wire distances (CuBe).

Fig. 5.1 shows 100 wire distance measuring runs with CuBe wires. The peaks indicate that the wire distances were obtained for the repeated measurements of the same wires.

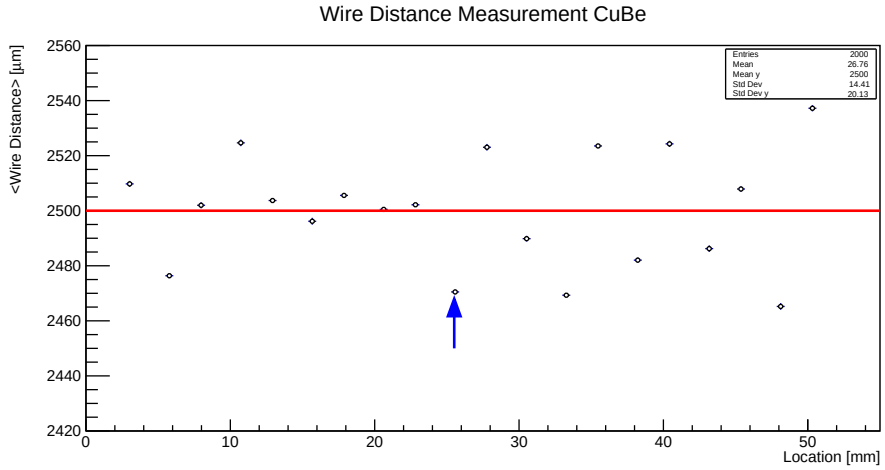

Figure 5.2: Mean wire distance measurement of each CuBe wire, plotted as a function of the wire location. The blue arrow marks the distance between wire 10 and 11 (see Fig. 5.3). The red line is a fit indicating the mean value over all wires.

In Fig. 5.2, the mean distance between two wires is plotted as a function of their location. The vertical error bars in the size of the points showing the one-sigma standard deviation. The fact that these error bars are so small is an indication of the reproducibility and thus of the quality of the individual measurements.

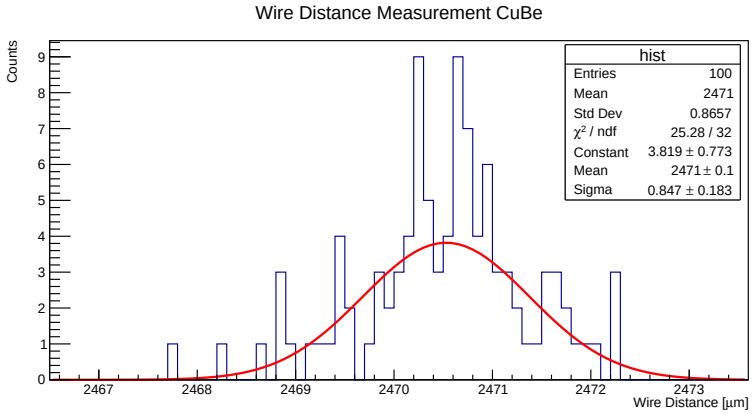

Figure 5.3: 100 measurements of the distance between wire 10 and 11 (CuBe).

In Fig. 5.3 the wire distance measurement between wire number 10 and 11 is shown. A Gaussian fit was applied with a standard deviation of $\sigma = 0.847$ µm. From this it can be shown that the new setup can reproduce the distance measurement on the order of few µm.

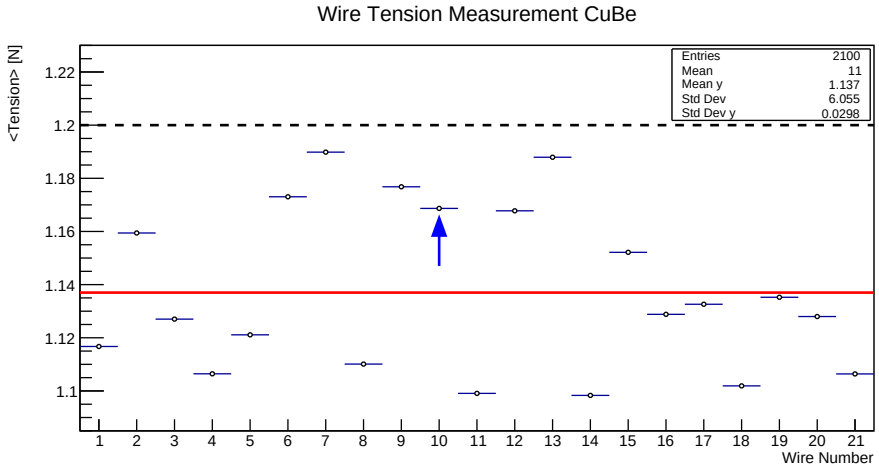

Figure 5.4: The mean wire tension of all 21 CuBe wires. The blue arrow marks wire number 10 from Fig. 5.5 The dashed line is the target value and the red line is the mean value.

In Fig. 5.4, the mean tension of the individual CuBe wires is plotted as a function of the wire number. Here, the vertical error bars are smaller than the size of the points and showing the standard deviation of the 100 entries per bin.

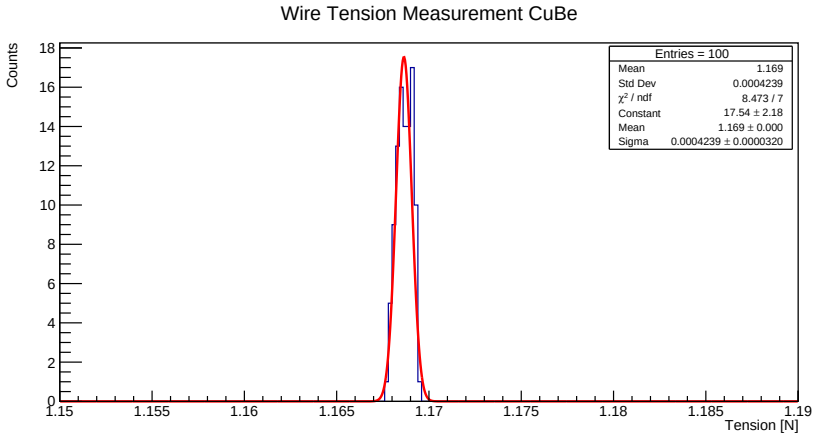

Figure 5.5: The tension measurement of the 10th wire in the CuBe measurement run.

In Fig. 5.5 the tension of wire number 10 is shown. A Gaussian fit is plotted on top with a standard deviation of $\sigma = 0.0004$ N. From this result, it can be seen that the new setup measures very accurately.

5.2 AuW-Wire

For the AuW measurement, a frame winding was wound with a target value of 0.5 N for the wire tension and 2.5 mm for the wire distances. After the winding, the 21st wire was streched mechanically to check if the software can detect the manipulated wire. During the production of a TRD chamber, it could happen that wires get streched when touched. This manipulation should simulate this.

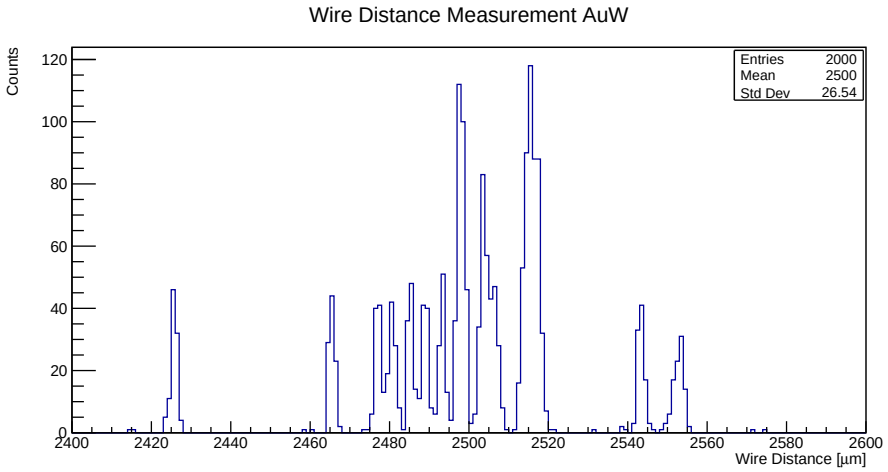

Figure 5.6: 100 distance measurements of two consecutive wires for all 20 wire distances (AuW).

Fig. 5.6 shows 100 wire distance measurements with AuW wires. In contrast to the CuBe measurement, individual outliers can be seen. Due to their smaller radius AuW wires are more sensitive to the slightest vibration.

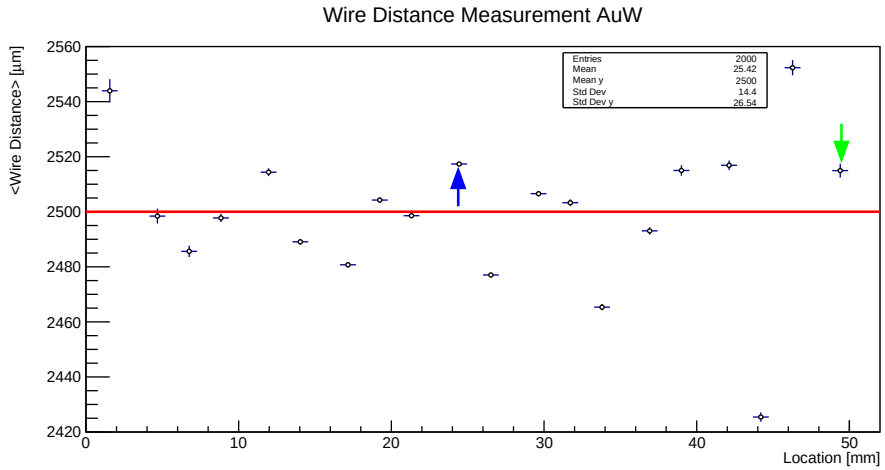

Figure 5.7: Mean wire distance of all individual AuW wires as a function of their location. The blue arrow marks the wire distance between wire 10 and 11 (see Fig. 5.8) and the green arrow marks the wire distance between wire number 20 and 21 (see Fig. 5.9), which is the distance to the manipulated wire. The red line is a fit indicating the mean value over all wires.

In Fig. 5.7, the mean wire distances were plotted as a function of their locations. Note that distance 18 is small and distance 19 is very large. This shows that wire number 19 is shifted to the left. The vertical error bars are barely visible. Similar to the CuBe measurement this fact indicates the reproducibility and thus of the quality of the individual measurements.

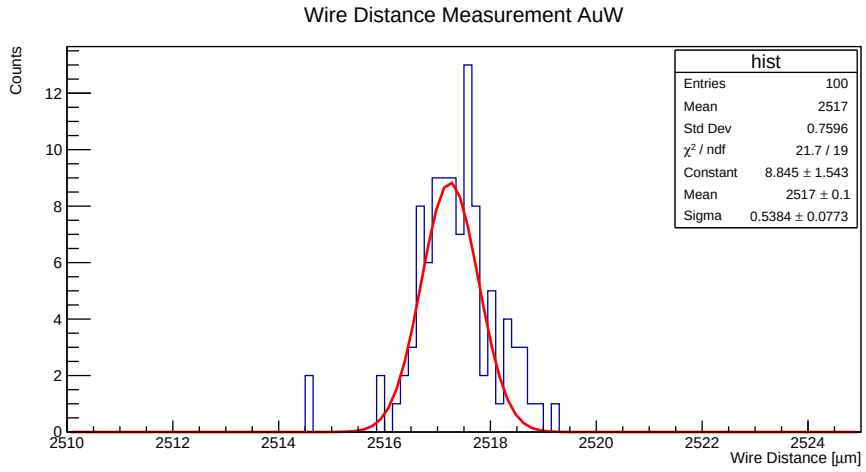

Figure 5.8: 100 measurements of the distance between wire 10 and 11 (AuW).

In Fig. 5.8 the wire distance measurement between wire number 10 and 11 is shown. A Gaussian fit was applied with a standard deviation of $\sigma = 0.54$ µm. From this result, it can be seen clearly that the new setup also measures very accurately with AuW wires.

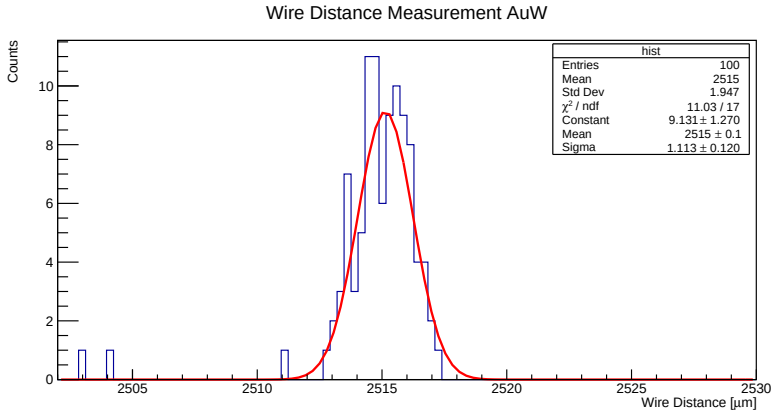

Figure 5.9: 100 measurements of the distance between wire 20 and 21 (AuW). In this case the distance to the manipulated wire.

In Fig. 5.9 the wire distance measurement between wire number 20 and 21 is shown. On this histogram a Gaussian fit was applied with a standard deviation of $\sigma = 1.11$ µm. The manipulation of the 21st wire may be responsible for the higher σ. Nevertheless, the result is accurate.

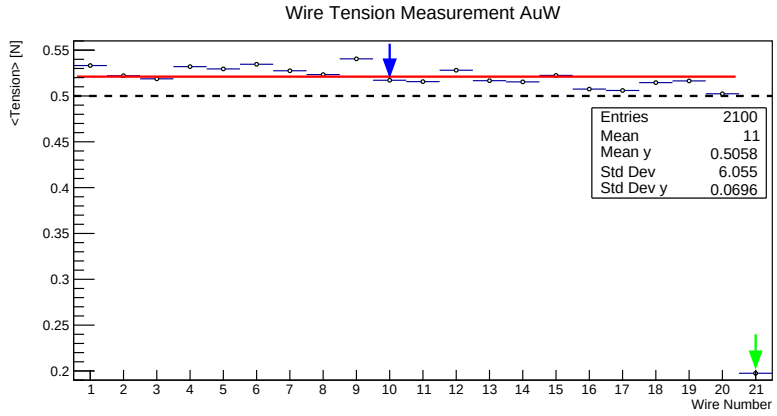

Figure 5.10: The mean wire tension of all 21 AuW wires. The blue arrow marks wire number 10 (see Fig. 5.11) and the green arrow the manipulated wire number 21 (see Fig. 5.12). The dashed line is the target value and the red line is the mean value (without consideration of the 21st wire).

In Fig. 5.10, the mean tension of the individual AuW wires is plotted as a function of the wire number. Again, the vertical error bars are smaller than the size of the points and showing the standard deviation of the 100 entries per bin. The drop in the wire tension of wire number 21 is clearly visible. The manipulation of the wire had a significant effect and can be reliably measured with the new setup.

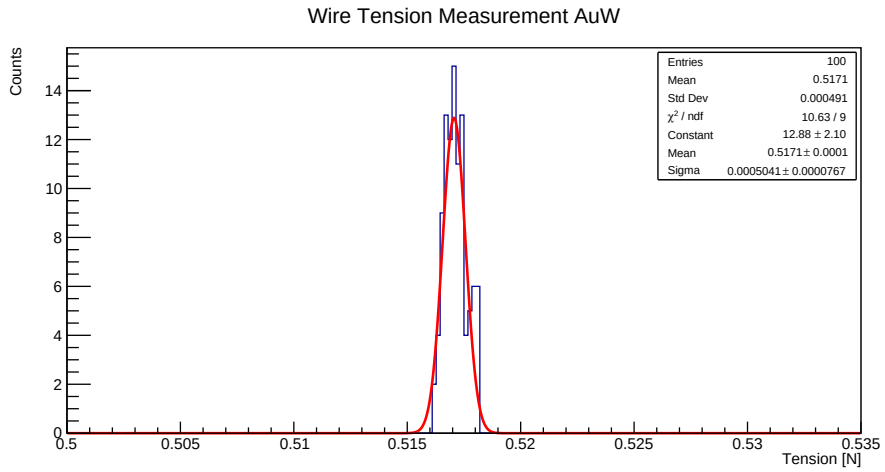

Figure 5.11: 100 tension measurements of wire number 10 (AuW).

In Fig. 5.11 the tension of wire number 10 was investigated. A Gaussian fit was applied with a standard deviation of $\sigma = 0.0005$ N. From this result it can be seen that the new setup also measures very accurately with the wire tension.

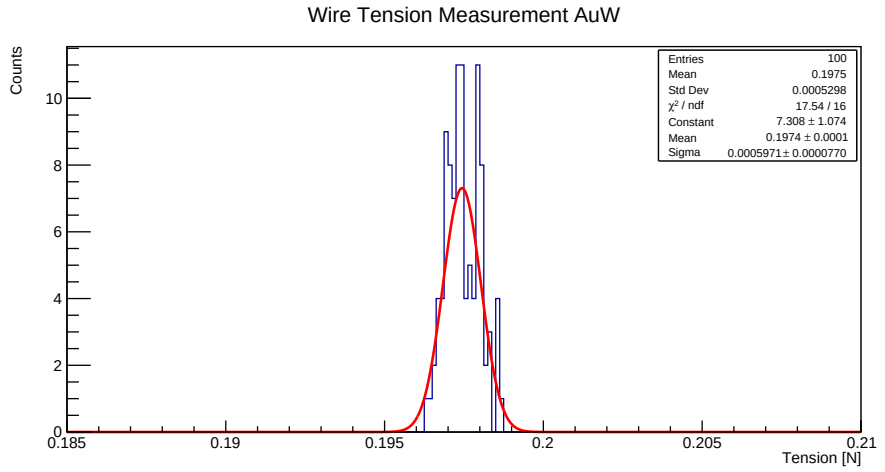

Figure 5.12: 100 tension measurements of wire number 21 (AuW).

In Fig. 5.12 the tension of wire number 21 is shown. On this histogram a Gaussian fit was applied with a standard deviation of $\sigma = 0.0006$ N. Again, the new setup measures very accurately.

5.3 CuBe-Wire (non-optimal)

As another example, a faulty winding with 392 wires and 391 distances was investigated. Also for this CuBe measurement, a frame winding was generated with a target value of 1.2 N for the wire tension and a target value of 2.5 mm for the wire distances. The parameters for the winding software were correctly set, but a brake on the winding machine was accidentally tightened. This configuration should allow to study the effects of careless work with the winding machine. Only one run was performed for this case.

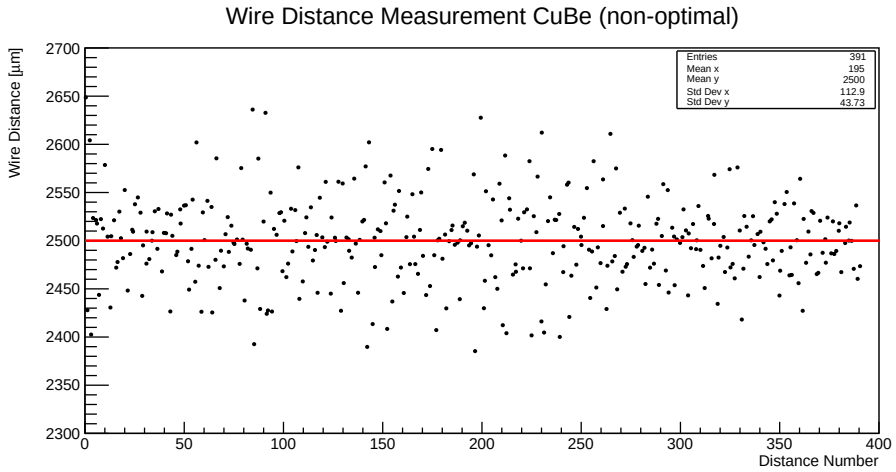

Figure 5.13: Plot of the wire distance measurement of the non-optimal CuBe winding. The red line is the mean value.

Based on the measured values, it can be seen that the brake on the winding machine had a small effect on the wire distances. The measured values vary slightly more than expected (see Fig. 5.13).

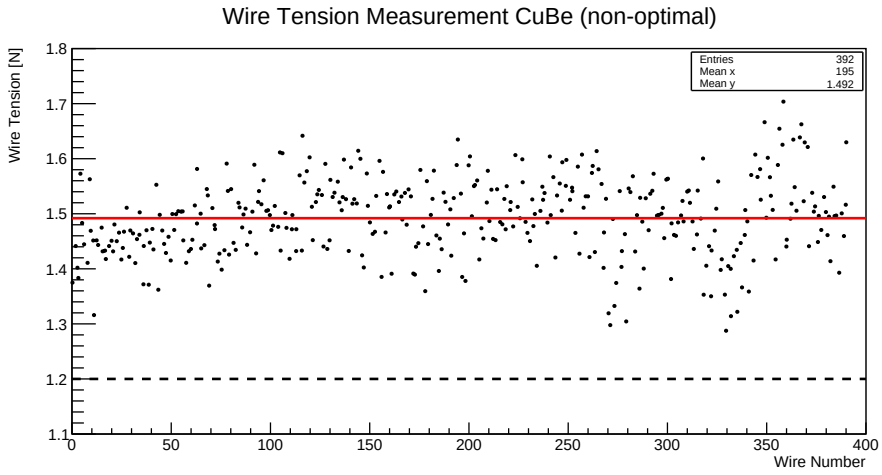

Figure 5.14: Plot of the wire tension measurement of the non-optimal CuBe winding. The dashed line is the target value and the red line is the mean value.

As with the wire distance measurement, the tightened brake had an effect on the wire tension measurement (see Fig. 5.14). The tightened brake caused more friction. As a result, the winding machine had to exert more force. Not only that the wires were too tight, you can see from the measurement results strong fluctuations. Deviations of up to 0.49 N to the target value of 1.2 N could be observed. Based on these measurements, this winding would be unusable for further processing.

6 Summary and Outlook

In this bachelor thesis, the problems of the WTD were documented and solutions sought. It was achieved that only with rewritten software the measuring process is faster, more efficient and quieter in the performance, without an installation of new hardware, which would have been associated with acquisition costs. In addition, the new software is easier to use and ready for future upgrades.
If the rounded-up double sigma values are used, the precision of the wire distance measurement for CuBe and AuW is ± 2 µm. Wire tensions of CuBe and AuW wires can be measured with a precision of ± 1 mN. The newly developed software thus offers sufficient measurement accuracy for the quality control of TRD production.
If more precise measurement results and less maintenance are required, then the mechanics of the WTD must be updated. Instead of the thread a linear guide could be installed. The POSIMAG length measuring system could be replaced with an optical measuring method. The software could be modified in the future to detect and measure alternating wire types.
In addition, care should be taken during a measurement to avoid vibrations. In particular, AuW wires are very sensitive to vibration, which can negatively affect the measurements.
Finally, it should be mentioned that mindfulness is required during handling of the wires. Minor flaws can have serious effects for the performance of the TRD chambers.

7 Attachments

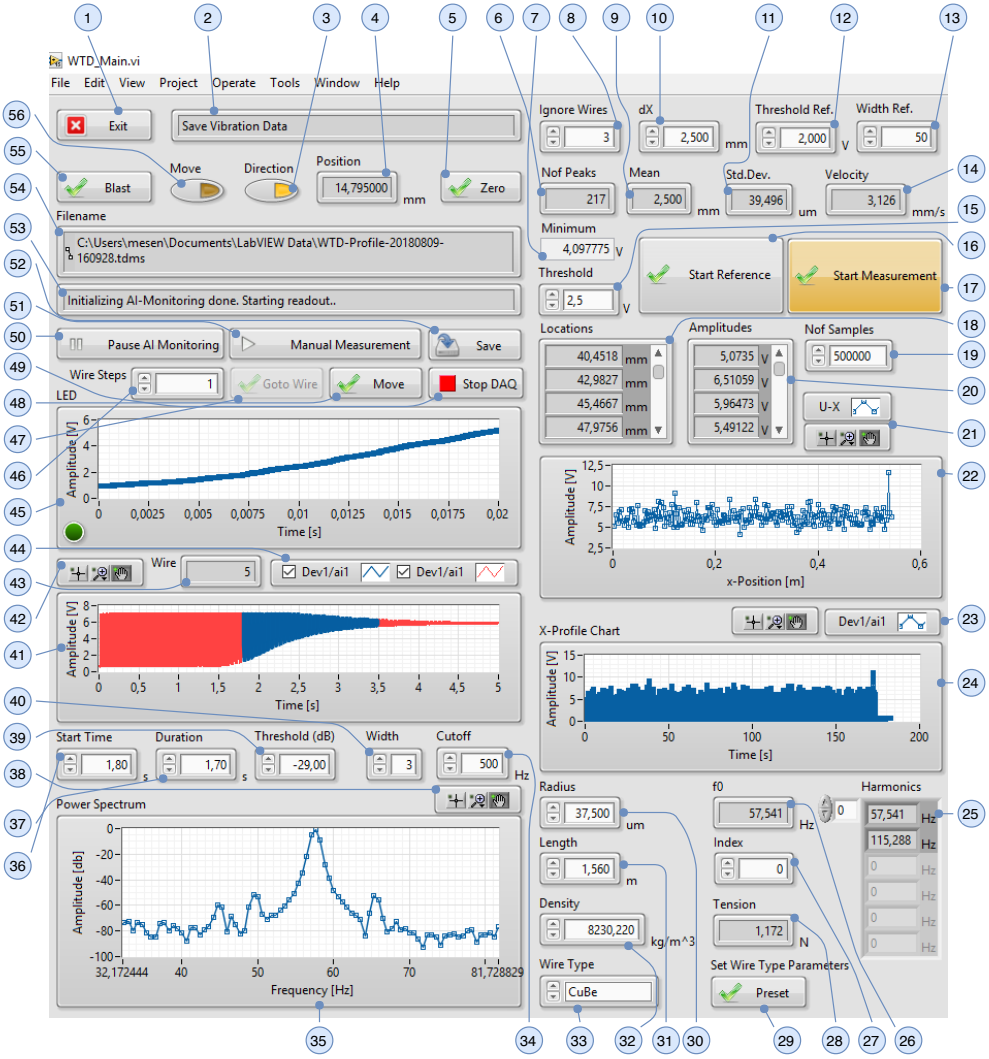

Figure 7.1: Screenshot of the Graphical User Interface. All functions are numbererd.

1. **Exit:** Quits the program.
2. **Message Handling Loop Status Bar:** Shows the current status of the Message Handling Loop.
3. **Direction:** Changes the direction of movement of the probe carrier.
   on = right, off = left
4. **Position (mm):** Displays the current position of the probe carrier in mm.
5. **Zero:** Resets the position value to 0.
6. **Nof Peaks:** Displays the number of wires the Peakfinder has found.
7. **Minimum:** This indicator displays the minimum value of all found reflection amplitudes from the "Amplitudes" index.
8. **Ignore wires:** This function ignores a certain number of wires in the reference run. Since the wires are not optimally tensioned at the beginning, as well as at the end of a winding, this function can be used if necessary.
9. **Mean (mm):** Shows the average value of the measured wire distances.
10. **dX (mm):** This tab specifies the setpoint of the distance of the wires. This setting is processed for the "Mean" indicator.
11. **Std.Dev. (µm):** Displays the calculated standard deviation of the measured wire distances.
12. **Threshold Ref. (V):** Specifies the threshold for the Peak Detector.VI in the reference run.
13. **Width Ref.:** Specifies how many measurement points should be set to the amplitude. This setting is flows into the Peak Detector.VI in the reference run.
14. **Velocity (mm/s):** Indicates the speed of the probe carrier calculated by the Peakfinder.
15. **Threshold (V):** After the reference run, the software proposes a threshold in this input box. Specifies the threshold for the Check for Leading Edge.VI in the wire tension measurement.
16. **Start Reference:** Starts the reference run program. Measures reflections, distances and the number of wires found.
17. **Start Measurement:** This function is the heart of the program. Measures the wire tensions.
18. **Locations (mm):** All found wires are displayed in the location index together with their position.

19. **Nof Samples:** Specifies the number of samples recorded for the wire tension measurement at a sample rate of 100000 Hz, for each wire. This setting should match the upper limit of the subset.
20. **Amplitudes (V):** The corresponding reflection amplitudes are shown in this index.
21. **Chart tools:** Tools to enlarge, reduce the size of a section of the chart etc.
22. **Wire position chart:** Similar to the LED chart, the measured amplitudes are shown. In this case, the voltage amplitude is plotted on the position in meter.
23. **Chart tools:** Tools to enlarge, reduce the size of a section of the chart etc.
24. **X-Profile chart:** Exact to the LED chart, the measured amplitudes are shown. In this case, the whole spectrum is shown.
25. **Harmonics (Hz):** Shows the calculated harmonic vibrations of the subset using the Fourier transform.
26. **f0 (Hz):** Displays the measured fundamental harmonic.
27. **Frequency Index:** This index lets you manually select a frequency from the "Harmonics" index.
28. **Tension:** This indicator displays the measured wire tension.
29. **Set Wire Type Parameter:** Activates the values from the Wire Type input field.
30. **Radius (µm):** This input field determines the radius of the wire. This value can be found on the label of the spool.
31. **Length (m):** This input field determines the length of the wire, which is stretched over the frame or the chamber.
32. **Density (kg/m^3):** This input field determines the density of the wire. This value can be found on the label of the spool.
33. **Wire Type:** This input field contains predefined wire parameters.
34. **Cutoff:** In the power spectrum the dominant harmonic vibration is searched. Cutoff is the maximum frequency that should be considered.
35. **Power Spectrum chart:** This chart plots the result of the Fourier transform.
36. **Start Time (s):** Defines the range of the start time for the Fourier transform.
37. **Duration (s):** This value defines the range of the duration to which Fourier transform is applied.

38. **Chart tools:** Tools to enlarge, reduce the size of a section of the chart etc.
39. **Threshold (dB):** This value defines the threshold value for the Peak Detector.VI in the wire tension measurement.
40. **Width:** Specifies how many measurement points should be set to the amplitude. This setting is for the Peak Detector.VI in the wire tension measurement.
41. **Subset chart:** This chart shows the subset of the vibration of the wires. The subset can be changed using the associated parameters for better results. For better illustration, the subset is highlighted in color.
42. **Chart tools:** Tools to enlarge, reduce the size of a section of the chart etc.
43. **Wire:** This indicator displays the current wire number.
44. **Marker:** Tool to change the appearance of the chart.
45. **LED chart:** The LED chart shows the current reflection that the sensor measures. The amplitude in volts is plotted on the time in seconds.
46. **Wire Steps:** With this tab a specific wire can be approached.
47. **Goto Wire:** This button was designed to drive wires in both directions. Due to the missing of the POSIMAG length measuring system, it is not possible to use this function. The program is however finished and can be used in a future upgrade of the WTD.
48. **Move:** In combination with the "Wire Steps" index, it is possible to approach a wire.
49. **Stop DAQ:** Stops all data acquisitions.
50. **Pause AI-monitoring:** Pauses the analog input.
51. **Manual Measurement:** This function is intended for a subsequent measurement of a wire. Gives the command to move to the next wire and measure the vibration.
52. **Save:** Saves the data of a manual measurement.
53. **DAQ Status Bar:** Shows the current status of the DAQ card.
54. **Filename:** This bar displays the current file path.
55. **Blast:** Initiates an air blast. This allows the compressed air supply to be checked.
56. **Move:** Initiates the movement of the probe carrier.

Bibliography

[1] Etienne Bechtel. *Electron identification with a likelihood method and measurements of di-electrons for the CBM-TRD*. Masterthesis. Johann-Wolfgang Goethe Universität Frankfurt Institut für Kernphysik Frankfurt am Main, 2017.

[2] Norbert Pucker Christian B. Lang. *Mathematische Methoden in der Physik*. Spektrum Lehrbuch Verlag, 2016.

[3] Goodfellow GmBH. *Manufacturer's note on the label of the coil*.

[4] Holger Gottschlag. *Entwicklung einer Apparatur zur automatisierten Positions- und Spannungsbestimmung von Drähten in Vieldrahtproportionalkammern*. Diplomarbeit. Westfälische Wilhelms-Universität Münster Institut für Kernphysik Münster, 2005.

[5] The TRD Working Group. *Technical Design Report for the CBM*. 2017.

[6] *Großes Handbuch technische Formeln*. Buch und Zeit Verlag, 1998.

[7] National Instruments. *Benefits of Programming Graphically in NI LabVIEW*. 2013. URL: http://www.ni.com/white-paper/14556/en/.

[8] National Instruments. *National Instruments PCIe-6320 DAQ card*. 2016. URL: http://www.ni.com/en-us/support/model.pcie-6320.html.

[9] National Instruments. *Peak Detector VI*. 2012. URL: http://www.ni.com/white-paper/3770/en/.

[10] National Instruments. *Queued Message Handler Template documentation*. 2017. URL: http://www.ni.com/tutorial/53391/en/.

[11] National Instruments. *Queued Message Handler Template picture*. 2017. URL: http://www.ni.com/cms/images/devzone/tut/loc_queued_message_handler_20120627135111.gif.

[12] National Instruments. *The NI TDMS File Format*. 2017. URL: http://www.ni.com/white-paper/3727/en/.

[13] National Instruments. *What Is DIAdem?* 2018. URL: http://www.ni.com/en-us/shop/data-acquisition-and-control/application-software-for-data-acquisition-and-control-category/what-is-diadem.html.

[14] National Instruments. *What is NI-DAQmx?* 2018. URL: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000P8baSAC&l=en-US.

[15] Paul Tipler. *Physik*. Spektrum Lehrbuch Verlag, 1998.

[16] Wikipedia contributors. *Drahtkammer*. 2015. URL: https://de.wikipedia.org/w/index.php?title=Drahtkammer&oldid=143844192.

[17] Wikipedia contributors. *Facility for Antiproton and Ion Research*. 2018. URL: https://en.wikipedia.org/w/index.php?title=Facility_for_Antiproton_and_Ion_Research&oldid=826476311.

[18] Wikipedia contributors. *LabVIEW*. 2018. URL: https://en.wikipedia.org/w/index.php?title=LabVIEW&oldid=865637068.

[19] Wikipedia contributors. *Standard Model*. 2018. URL: https://en.wikipedia.org/w/index.php?title=Standard_Model&oldid=865408094.

[20] Wikipedia contributors. *Standard model of elementary particles*. 2006. URL: https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg.

[21] cbm wiki.gsi.de. *Sketch of the CBM detector SIS 100*. 2016. URL: https://cbm-wiki.gsi.de/foswiki/pub/PwgC2F/PwgModelsFigures/2016.03.14_CBM_Hades_SIS_100_cut.jpg.

8 Eigenständigkeitserklärung

Hiermit bestätige ich, dass ich diese Arbeit selbständig verfasst und keine anderen, als die angegebenen Quellen und Hilfsmittel verwendet habe.

Frankfurt am Main, 15.11.2018

Murat Esen