



Validation and study of the design and simulation model of the ProtoDUNE cryostats

Strain Gauge Analysis

CERN Summer Student Programme 2019

CERN Summer Student Report

Lars Bathe-Peters¹

Supervised by: Olga Beltramello, Andrea Zani , Alberto Rigamonti
(CERN)

20th September, 2019

Abstract

A strain gauge analysis on the DP-ProtoDUNE (NP02) cryostat was performed in order to validate the detector design and dimensions. Due to the positive outcome of this and further analyses a robust and predictable structural behaviour of the ProtoDUNE cryostat can be confirmed. A pressure test yielded no danger of weakening the cryostat structure, which verifies the correct design and dimensions, allows the improvement of current predictions of FEA models and safety capabilities. Within this Summer Student Project a generic plotting toolkit was created in order to generate monitoring means for a stable and automated monitoring of data. Due to the generality of the code, this toolkit holds the potential to be adapted to an outreach beyond the ProtoDUNE experiment.

¹mailto:lars_bathe-peters@fas.harvard.edu or lars.bathe-peters@cern.ch or lars@physik.tu-berlin.de

Contents

1 The DUNE experiment	3
2 ProtoDUNE	4
3 ProtoDUNE LArTPC detector technologies	5
3.1 Single-Phase configuration	6
3.2 Dual-Phase configuration	6
4 Cryostat	7
5 Cryostat Deformation Analysis	8
6 Strain Gauge Analysis	8
6.1 Tools for the Strain Gauge Analysis	9
6.2 Corrections to the data	9
6.2.1 Zeroing	10
6.2.2 Moving average correction	11
6.2.3 Overpressure correction	11
6.2.4 Temperature correction	12
6.3 Strain vs. Liquid Height	14
7 Pressure Test	14
8 Conclusion and Outlook	17
9 Analysis code	22
10 Extract data from database (RetrieveOracleDBdata.py)	22
10.1 Specify start and end date	22
11 Get number of values per sensor (get_column_length.C)	23
12 Save values to .root-file (create_variable_file.C(int x))	24
13 Apply corrections to data (make_variable_corrections.C)	24
14 Merge file_* .root-files (merge_root_files.C)	25
15 Generate plots (make_plots.C)	25
16 Run all scripts (run_it.sh)	27
17 Future perspective of this work	28
18 Appendix	28

List of Figures

1 LBNF / DUNE project	4
---------------------------------	---

2	ProtoDUNE in EHN1	5
3	Novel LArTPC detector technologies	6
4	DP-ProtoDUNE (NP02) cryostat	7
5	Strain gauge	9
6	Example for data without corrections	10
7	Example for zeroed data	10
8	Example for zeroed, moving averaged data	11
9	Example for zeroed, moving averaged, pressure corrected data	12
10	Calculation of the pressure coefficient.	12
11	Temperature sensor TT_ambiance_Top	13
12	Example for zeroed, moving averaged, pressure and temperature corrected data	13
13	Strain vs. Liquid Height	14
14	Pressure development at the 2nd NP02 pressure teston 12 th August 2019	15
15	Strain development at the 2nd NP02 pressure teston 12 th August 2019.	16
16	Strain development at the 2nd NP02 pressure teston 12 th August 2019.	17
17	start and end in RetrieveOracleDBdata.ipynb	22
18	Example for data_input.txt	23
19	Example for column_count.root	23
20	Example for root_files/file_* .root	24
21	Example for root_files/file_* .root with the added TBranch called corrections	25
22	merged_file.root contains the merged TTrees data and corrections .	25
23	plots.root containing chosen sensors of NP02	26
24	Example plots for sensors installed on the ProtoDUNE Dual-Phase detector cryostat	27

1 The DUNE experiment

The Deep Underground Neutrino Experiment (DUNE) is a next-generation long-baseline neutrino-oscillation experiment. The operation is estimated to start by 2026 at Fermilab. It will consist of a near detector to be located at a distance of 575 m from the neutrino source at Fermilab in Illinois and a far detector (FD) located 1.5 km underground at the Sanford Underground Research Facility (SURF) in South Dakota (USA) at a distance of 1300 km from the origin of the neutrino beam produced at Fermilab. The Long-Baseline Neutrino Facility (LBNF) provides the infrastructure for DUNE.

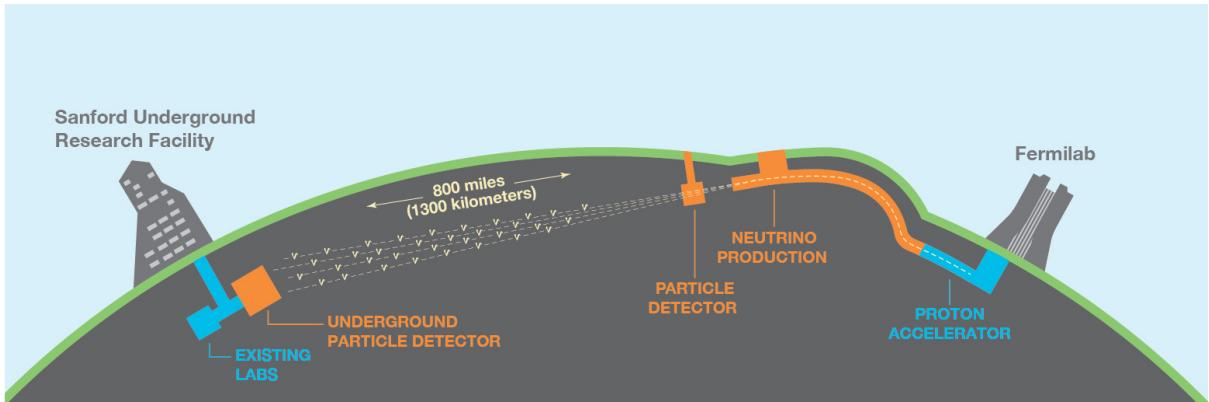


Figure 1: LBNF / DUNE project.

The intensity and energy spectrum of the incoming neutrino beam will be characterised by the near detector. The primary science goals of DUNE are to

- perform neutrino oscillation measurements, i. e.
 - Measurement of the charge parity (CP) phase δ_{CP}
 - Determination of the neutrino mass ordering to solve the neutrino mass hierarchy problem
 - Measurement of the mixing angle θ_{23} and determination of the octant in which it lies
 - Sensitive tests of the three-neutrino paradigm in order to infer about the possible existence of sterile neutrinos
- search for proton decay in various decay modes motivated by being able to get closer to a grand unification of forces,
- detect and measure a supernova ν_e flux to learn about core-collaps and black holes.

Some of DUNE’s ancillary goals include measurements of other accelerator-based neutrino flavor transition measurements, neutrino cross sections, studies of nuclear effects, and searches for dark matter.

The installation of the first module of the DUNE Far Detector FD is expected to start in 2022. The DUNE FD will consist of four similar Liquid Argon Time Projection Chambers (LArTPCs) with two different LArTPC technologies with a total fiducial volume of 40 kton. Each FD detector will be embedded in a cryostat with internal dimensions of 14.0 m (w) x 14.1 m (h) x 62.0 m (l). In order to test detector components, validate installation procedures, validate the detector designs and performance of the DUNE FD and collect test beam data, DUNE is prototyping two LArTPC technologies. The set of a Single-Phase (SP - NP04) and Dual-Phase (DP - NP02) LArTPC detectors is referred to ProtoDUNE. [1]

2 ProtoDUNE

ProtoDUNE is a set of two LArTPC detectors each holding approximately 800 t of liquid argon and located in Experimental Hall North 1 (EHN1) at CERN. In order to prepare

the operation of the largest LArTPC detectors in the world, the ProtoDUNE detectors are build at the CERN Neutrino Platform in order to characterize and calibrate the detector response among other reasons mentioned in section (1).

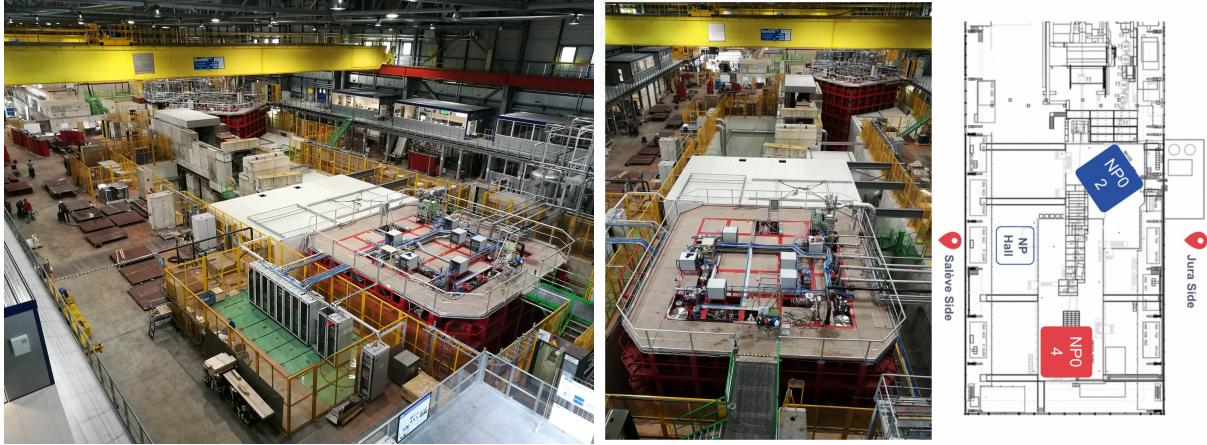


Figure 2: ProtoDUNE in EHN1.

Low-energy charged particles in the momentum range from 0.4 to $12 \frac{\text{GeV}}{\text{c}}$ are provided by extensions to the Super Proton Synchrotron (SPS) North Area secondary beam lines called H2 and H4 and respectively reach the DP- and SP-ProtoDUNE detectors. [2] The exposition of the DP- and SP-ProtoDUNE detectors to two independent low energy beam lines, allows the study and characterization of the LArTPC response to charged particles in the same energy range (500 MeV up to a few GeV) of prospective neutrino interactions in DUNE. Further purposes of ProtoDUNE are the validation of the novel technologies of the DP- and SP- detector technologies, the validation of the membrane cryostat technology and associated cryogenics, a test of the DUNE FD engineering solutions and installation procedures, to probe networking and computing infrastructure in order to handle data, optimize event reconstruction algorithms and quantify and reduce systematic uncertainties for the DUNE FD detector. [4]

3 ProtoDUNE LArTPC detector technologies

The ProtoDUNE LArTPC detectors are based on common TPC technologies. As a charged particle² traverses the detector medium, it scatters off and excites close enough atoms by kicking off ionisation electrons of these atoms. These ionisation electrons are drifted by an electric field towards the anode plane that consists of multiple planes of sense wires. After the ionisation electrons deposit charge on those wires, the corresponding measured waveforms of the wires can be used to reconstruct a three-dimensional image of the charged particle's track inside the TPC. As the excited argon dimers deexcite, they isotropically emit photons in the form of scintillation light by radiative decay. The drift direction of the particle's track can then be obtained by comparing the time of the flash of scintillation light (caused by a neutrino interaction) registered by the ARAPUCA light collection system or photomultiplier tubes (PMTs) with the time of the charge deposit on the wires. In addition, the scintillation light is used in order to identify a particle track inside the TPC. Hereby a simulation of the number of Photo Electrons (PE) per

²This charged particle can be a final state particle of a neutrino-nucleus interaction inside the LArTPC.

light detection device (track-to-light matching or Flash Matching (FM)) for each particle track is compared to the measured flash. On top of these basic operation principles, the DP- and SP-ProtoDUNE detectors inherit novel technologies that differ with respect to prior LArTPC detectors.

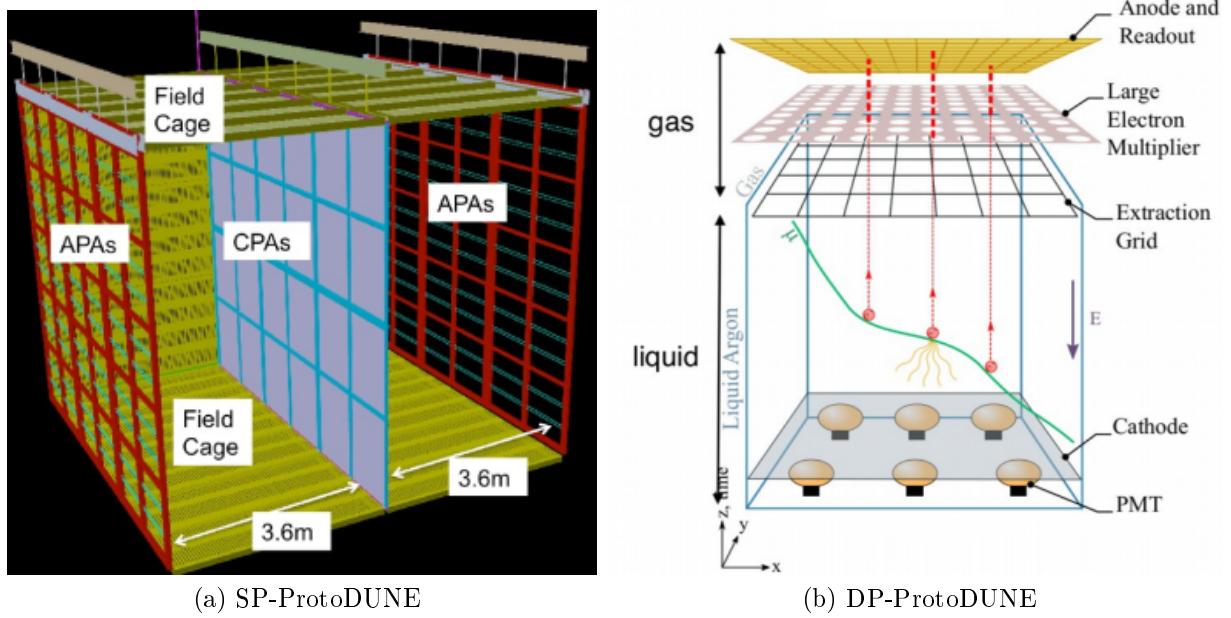


Figure 3: Novel LArTPC detector technologies.

Besides the APA and CPA, the remaining sides of the DP- and SP-ProtoDUNE detectors consist of the field cage. One novel technology of both the DP- and the SP-ProtoDUNE detector is a field cage made of aluminium. [4]

3.1 Single-Phase configuration

As consisting of two standard SP-LArTPCs with one anode plane assembly (APA) on each side (see fig. (3a)), the SP-ProtoDUNE LArTPC detector technology inherits common features with respect to prior LArTPC technologies. Novel technologies in SP-ProtoDUNE that differ with respect to existing neutrino detector technologies are the installation of a new device for liquid argon scintillation light detection called ARA-PUCA (Argon R&D Advanced Program at UniCAMP). Hereby a dichroic filter with two wavelength shifters traps scintillation light inside a high internal reflectivity cell in order to collect it with high efficiency. Another new feature is a Cathode Plane Assembly (CPA) made of a resistive material instead of conductive material. SP-ProtoDUNE was commissioned in 2018.

3.2 Dual-Phase configuration

As the name suggests the DP-ProtoDUNE detector makes use of two phases (liquid and gaseous) of argon (see fig. 3b) inside one DP-LArTPC and is a yet unprobed technology. Inside the DP-TPC a region of gaseous argon resides above the liquid phase. The ionisation electrons are accelerated by a strong electric field into the gaseous region by the

APA located on the top of the LArTPC near the cryostat. After migrating from argon in a liquid state to argon in a gaseous state there are four Charge Readout Planes (CRP) with 36 Large Electron Multipliers (LEMs) that amplify electrons and hence signals. The multi-layered APA collects charged particles such as electrons and provides spatial readout functionality. The light collection system consistent of PMTs are located underneath the detector. The main advantage of the DP-ProtoDUNE is an enhanced signal-to-noise ratio with respect to the SP-configuration. DP-ProtoDUNE was commisioned in July 2019 during my time as a summer student at CERN. [4]

4 Cryostat

The cryostat (see fig. 4) is required to store liquid argon at a temperature between 86.7 K and 87.7 K. Its cryostat dimensions and cryostat system designs are selected to be the same for both ProtoDUNE detectors. The cryostats surround the detectors consistent of APA, CPA and field cage. The ProtoDUNE cryostat consists of three layers. The inner cold membrane is made of a thin layer of stainless steel that is in contact with the liquid argon and prevents liquid leaks. Between the inner and outer layer (hull) there is a thick (80 cm) passive thermal insulation layer to intercept heat. The steel warm outer structure provides mechanical support for the membrane and its insulation and consists of vertical beams that alternate with a web of metal frames constructed to withstand the hydrostatic pressure of liquid argon and the pressure of gas volumes as well as to satisfy external constraints on the cryostat. There are several openings in the cryostat such as the beam entrance or the liquid argon supply. When the cryostat was under construction there was a Temporary Cryostat Opening (TCO) for detector assembly which is now closed. [3] [6]

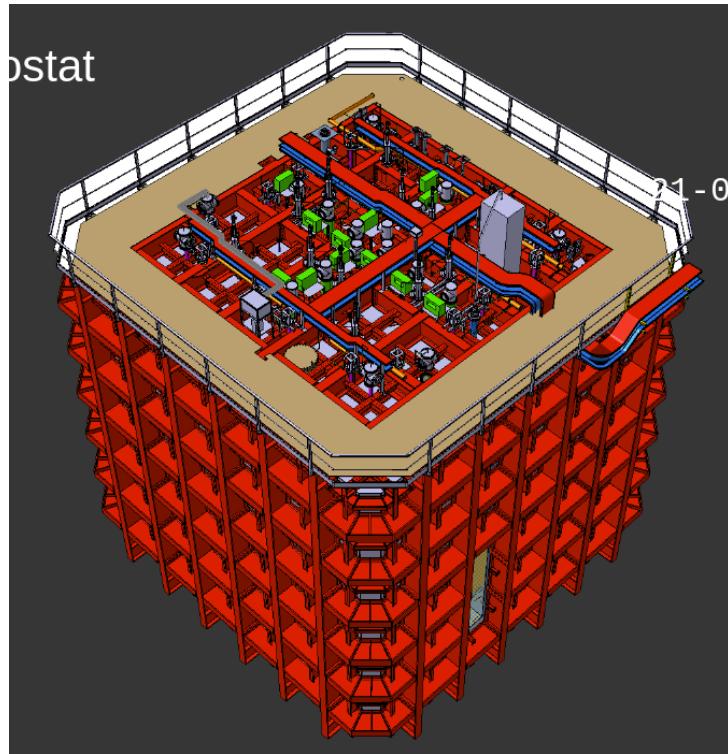


Figure 4: DP-ProtoDUNE (NP02) cryostat.

5 Cryostat Deformation Analysis

Sensors on the outer membrane of the cryostat measure mechanical deformations of this structure with the goal of validating design choices and simulations of the cryostat. These simulations are carried out by finite element analysis (FEA) models of the cryostate that aim to estimate structural deformations of the cryostat. The overall goal is to confirm the construction of a robust and predictable structural behaviour of the ProtoDUNE detector cryostat. This will hence validate the design and construction of the DUNE detector cryostat of a dimension that is approximately twenty times bigger than the ProtoDUNE cryostat and ensure its realization. The data produced in this analysis can be used to get feedback on the current model and allows a fine-tuning of current models to enhance simulation capabilities. As part of the cryostat deformation analysis, cryostat deformations were measured before, during and after the filling in order to research and learn about the structural behaviour and compare it with simulations that can possibly be improved for better predictions on prospective data. Furthermore the cryostat deformation analysis enhances safety capabilities. Another goal of this analysis is a stable and automatised monitoring of the data. Since an abnormal behaviour in terms of mechanical deformations of the cryostat can be recognised by deformation sensors, automatic monitoring is a tool to check the safety during the filling operation. There are 58 strain gauges, four temperature, two deformation sensors and several pressure sensor installed on the cryostat of NP02.

6 Strain Gauge Analysis

One measurable type of deformation is strain ε . Strain in the mechanical sense is the relative displacement between particles in a material body, i. e. the displacement between particles in a body relative to a reference length and therefore measured in units of microstrains. During deformations the metallic wire of a strain gauge as it can be seen in fig. 5³ is stretched or contracted, so that its electrical resistance changes with deformations and serves as a readout for the ratio of the change in length Δl and its original length l :

$$\varepsilon = \frac{\Delta l}{l} \quad \left[\frac{\mu m}{m} \right] \quad (1)$$

Positive strain means an elongation and negative strain means a contraction of the material. Strain gauges are used in Wheatstone Bridges to allow the measurement of strain in different directions.

Displacement is the difference in length between the linear distance before and after a deformation. In the case of the filling of NP02, the resulting overpressure leads to a positive displacement as the cryostat walls are expanding. In order to measure the displacement of the surface from its original position, laser light is reflected from a surface in order to measure the distance between displacement sensor and surface, which is then translated to the linear displacement of the material. There is one displacement sensor installed on each of the four sides of the NP02 cryostat.

In order to be able to see the influence of the filling on the structure of the cryostat

³Figures taken from <https://automationforum.co/strain-gauge-pressure-measurement/> respectively and <http://www.ni.com/en-us/innovations/white-papers/07/measuring-strain-with-strain-gages.html> respectively.

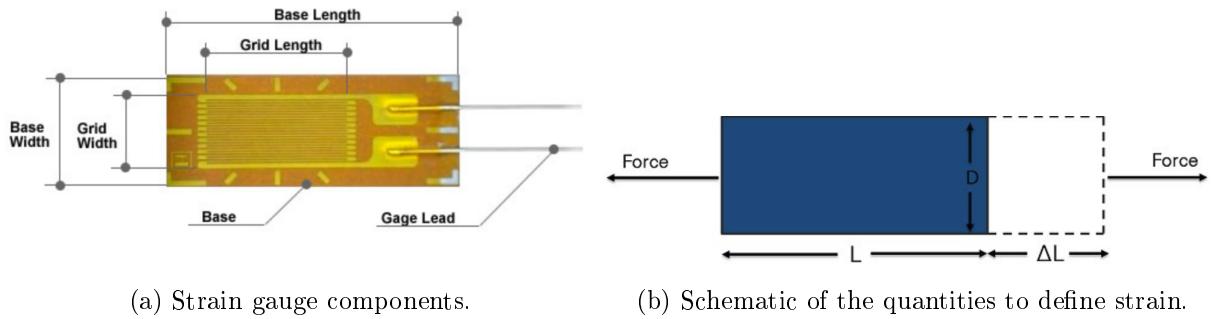


Figure 5: Strain gauge.

and compare data with simulations from model predictions, the readout of the sensors of NP02 is stored in the slow-control database ([6]) and has to undergo several corrections.

6.1 Tools for the Strain Gauge Analysis

As it is described in more details in the 'Technical details and documentation of the code' starting from page 21, the tools for a strain gauge analysis consist of a collection of scripts written in C++ (except for the data extraction script which is currently written in python, but can be written in any programming language of choice) using the software framework ROOT. At first data is extracted using a python script called `RetrieveOracleDBdata.ipynb`. This script extracts data from the NP02 or NP04 slow control database and writes data to a `.csv` – file. If the format in the `.csv` – file is the same, the scripts presented from page 21 can be used for any variables in any experiment as long as the data extraction script produces a `.csv` – file in the format needed to run the following scripts. After getting the number of measured values of every sensor with `get_column_length.C`, the script `create_variable_file.C` saves the data and other values of choice in a `.root`-file. This `.root`-file is then extended by correction values after running `make_variable_corrections.C`. After merging all output `.root`-files with `merge_root_files.C`, the datasets with applied corrections can be plotted using `make_plots.C`. There are different plot modes for different sensors available. All scripts can be run by executing `run_it.sh`.

6.2 Corrections to the data

In order to compare the data with simulations and correct environmental effects, the dataset consisting of i values ε_i has to undergo several corrections, e. g. zeroing, averaging, pressure and temperature corrections.

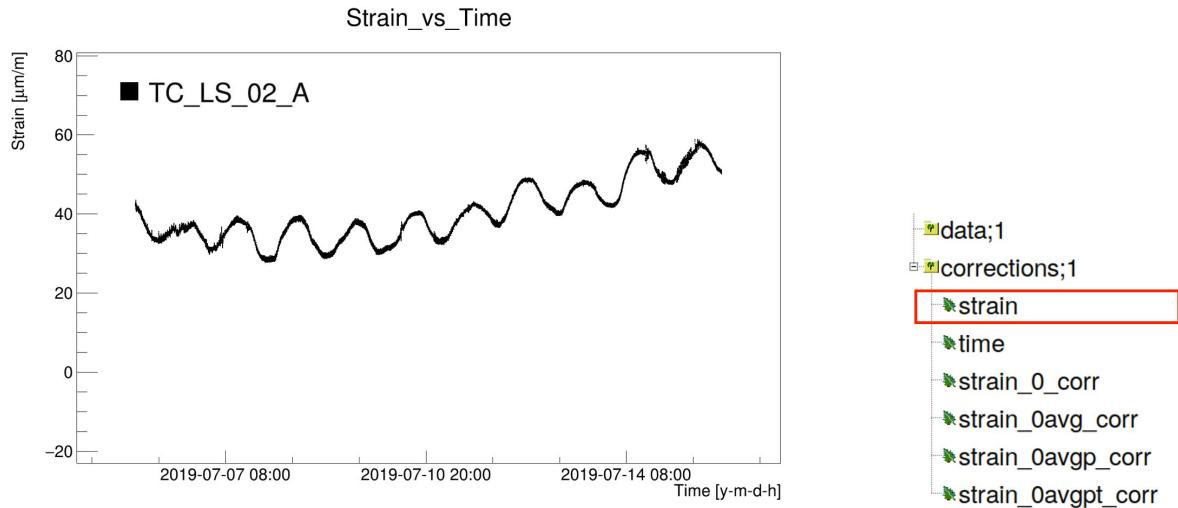


Figure 6: Example for data without corrections.

6.2.1 Zeroing

Before the start of the filling, the sensors installed on the walls of the cryostat took data and were not zeroed when the filling started. Therefore one has to account for variations in pressure, temperature, other environmental effects etc. To correct for drifts of the strain due to these effects (correct for drift and pressure effects before the filling), an offset is subtracted from the dataset:

$$\varepsilon_{0_i} = \varepsilon_i - \text{offset} \quad (2)$$

The zeroed dataset ε_0 can be seen in red in fig. 7.

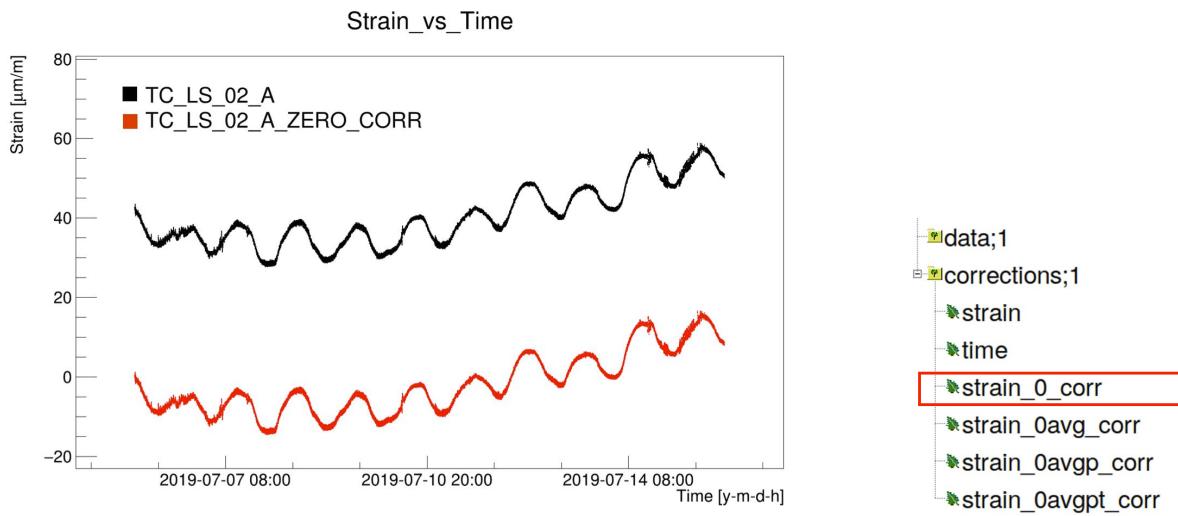


Figure 7: Example for zeroed data (red).

Zeroing is the removal of this offset in order to attain the real strain evolution due to the liquid only (and temperature variations) as the filling proceeds at constant overpressure. The effect of overpressure is added back afterwards in the overpressure correction step (see section 6.2.3).

6.2.2 Moving average correction

Due to the day and night cycle there are temperature fluctuations that impact the output of the deformation sensors. After zeroing, the data is moving average corrected in order to smooth out the temperature dependence of the deformation sensors arising from the day and night cycle (see fig. 8).

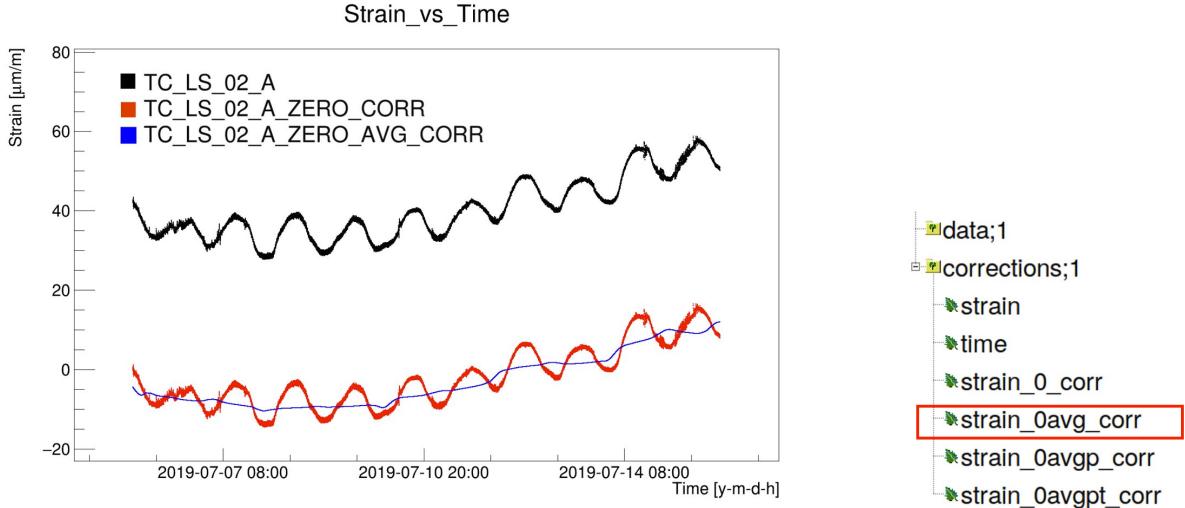


Figure 8: Example for zeroed, moving averaged data (blue).

For this the moving average of the values ε_{0_i} for the data corresponding to half a day at the beginning (first interval) and half a day before for the end of the data set (third interval) is calculated by substituting data points below the starting value and above the highest value of the dataset respectively by mirroring data points that lie within the dataset. The moving average of all other data points (second interval) in between the first and third interval are calculated by convoluting an array of one day length around a datapoint with an array filled with ones.

6.2.3 Overpressure correction

Due to the gaseous volume inside the cryostat there is an overpressure that induces another strain on the structure of the cryostat. In order to correct this overpressure effect and to compare the data with the simulation, the overpressure inside the cryostat (that was removed when zeroing (section 6.2.1)) is added back to the dataset and added to the zeroed, moving averaged corrected dataset:

$$\varepsilon_{0_{p_i}} = \varepsilon_{0_i} + \Delta\varepsilon_p \quad (3)$$

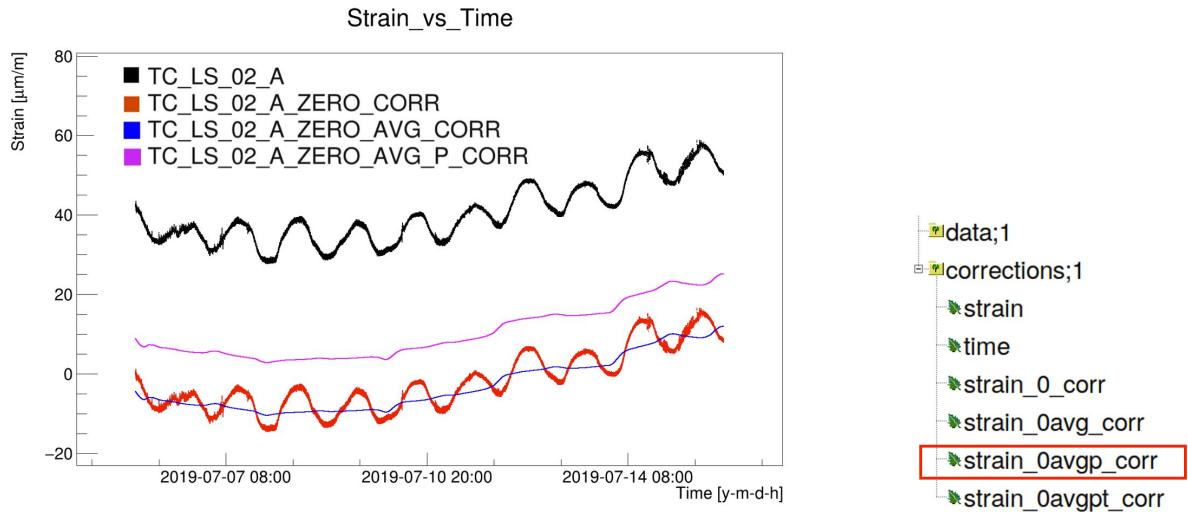


Figure 9: Example for zeroed, moving averaged, pressure corrected data.

The sensor dependent slope⁴ of the strain with respect to the overpressure is:

$$p_{\text{coeff}} = \frac{\Delta \varepsilon_p}{\Delta p} \quad \left[\frac{\text{microstrain}}{\text{mbar}} \right] \quad (4)$$

Pressure DP[MBAR]	Strain[um/m]
100	-6.46
150	-8.73
200	-9.12
250	-8.73
280	-6.42

Figure 10: Calculation of the pressure coefficient: The pressure coefficient p_{coeff} is the slope of the strain with respect to the pressure. The table on the right shows differential pressure values and the corresponding strain values for sensor TC_LS_02_A. From this p_{coeff} can be calculated by using equation 4.

For example the pressure coefficient for sensor TC_LS_02_A is $p_{\text{coeff}} = \frac{(-8.73 - (-6.46)) \frac{\mu m}{m}}{(150 - 100) \text{mbar}} = 0.045 \frac{\text{microstrain}}{\text{mbar}}$. A linear behaviour of the cryostat corresponds to $p_{\text{coeff}} = \text{const.}$ The pressure coefficient p_{coeff} is multiplied by the initial overpressure $\Delta p \equiv P_{\text{REF}}$ (pressure at the start of filling on 5th July 2019 at 18:10 was $P_{\text{REF}} = 23 \text{ mbar}$). Then the resulting difference in strain due to overpressure $\Delta \varepsilon$ is added to the zeroed, moving averaged data.

6.2.4 Temperature correction

In analogy to the overpressure correction, the dataset is corrected to account for temperature effects. Hereby the strain difference $\Delta \varepsilon_T$ per degree temperature (slope) $\Delta T_i = T_i - T_{\text{REF}}$ (the temperature at the start of the filling was $T_{\text{REF}} = 26^\circ \text{ C}$) is

$$T_{\text{coeff}} = \frac{\Delta \varepsilon_{T_i}}{\Delta T_i} \quad \left[\frac{\text{microstrain}}{^\circ \text{C}} \right] \quad (5)$$

⁴The p_{coeff} values for the sensors are stored in `P_coeff_NP02.txt`

and thus calculated in analogy to the pressure coefficient p_{coeff} . The T_{coeff} values for the sensors are stored in `T_coeff_NP02.txt`. Hereby the temperature set T is the moving averaged dataset of temperature sensor `TT_ambiance_Top` (see fig. 11).

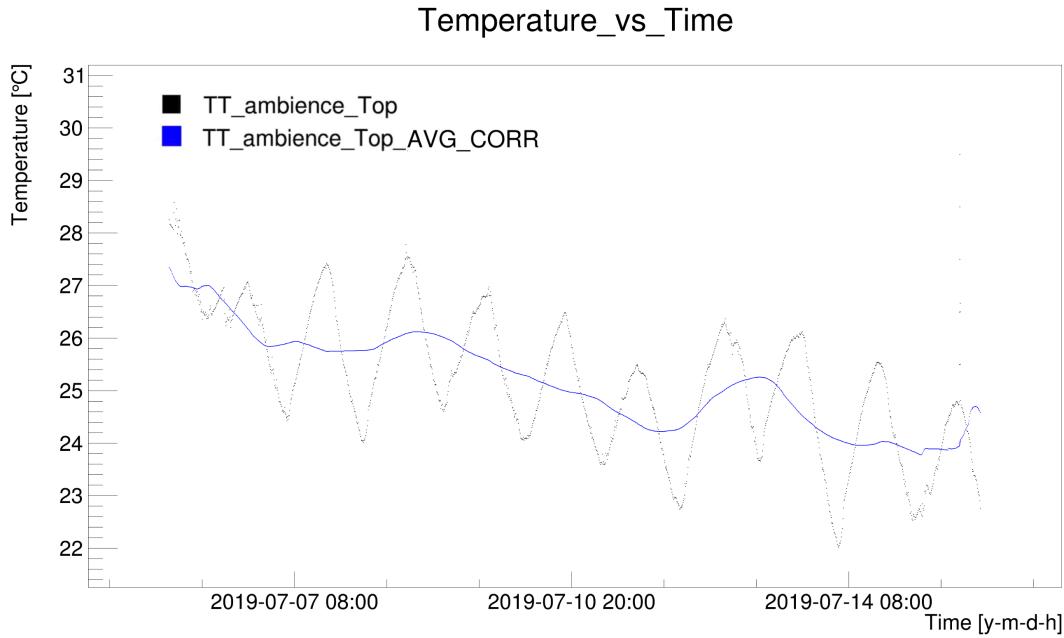


Figure 11: The zeroed, moving averaged data of temperature sensor `TT_ambiance_Top` is used for the temperature correction.

The resulting $\Delta\varepsilon_T$ is used to correct the zeroed, moving averaged, pressure corrected dataset in terms of temperature effects according to

$$\varepsilon_{0pt_i} = \varepsilon_{0p_i} + \Delta\varepsilon_{T_i} . \quad (6)$$

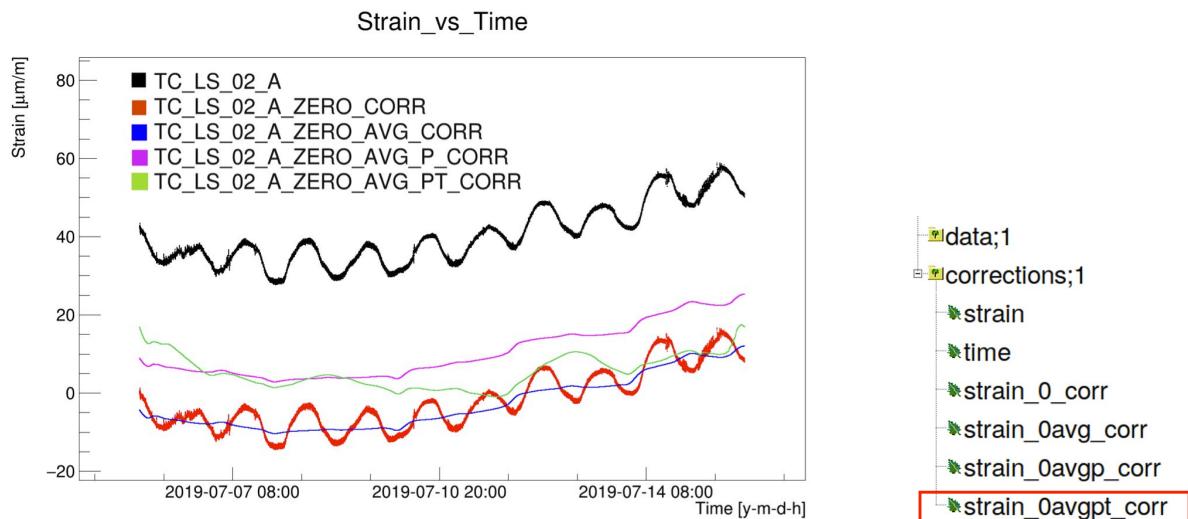


Figure 12: Example for zeroed, moving averaged, pressure and temperature corrected data.

6.3 Strain vs. Liquid Height

Fig. 13 shows the comparison between data and simulation for the strain with increasing liquid argon height during the filling of NP02. The corresponding strain gauge 4M_BS_VER_03 is located in the middle outer structure in a flat and smooth area.

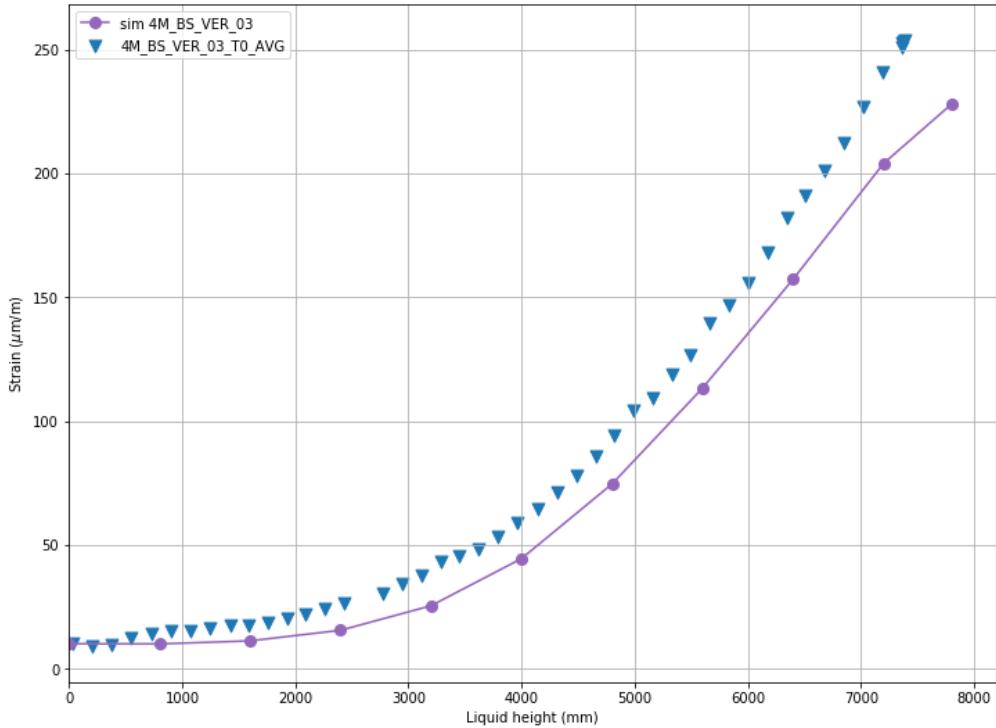


Figure 13: Strain vs. Liquid Height. purple: simulation, blue: data.

Fig. 13 agrees with the expectation of measuring an increasing strain with a rising liquid height. Overall an agreement between the simulation and data is observable. The simulation can still be improved for higher values of the liquid height.

7 Pressure Test

The purpose of the pressure test is to address safety issues of the cryostat. Hereby the pressure inside the detector is increased up to 280 mbar and then decreased again until the starting value (see fig. 14). There is one pressure test before the filling to confirm that the inner structure of the cryostat has no weakpoints and one pressure test at the end of the filling to assure that the filling has not modified the elastic behaviour of the structure of the cryostat due to the insertion of the liquid argon. The first pressure test involves the investigation of symmetry and nonlinear effects. The behaviour of the cryostat is referred to as symmetric if the cryostat remains in a symmetric shape when increasing pressure. Symmetry can be investigated by looking for a similar readout of sensors that are symmetrically located on the cryostat such as the strain gauges belonging to the plots in fig. 15. An asymmetric behaviour gives rise to nonlinear effects during the process of increasing pressure. In terms of nonlinearity the cryostat is expected to stay in an elastic range, i. e. the cryostat should deform with increasing pressure and return to its original shape when reaching the starting pressure. Nonlinearity can be

investigated by looking at p_{coeff} (see equation (4)). A linear behaviour is reflected by p_{coeff} being constant. The second pressure test confirms the outcome of the first pressure test in terms of symmetry and elastic behaviour and allows to verify that the structure is compliant to its safety requirements with respect to the safety valve⁵. The comparison between the plots shown in fig. 14 and 15 shows that pressure and strain behave nearly same over time. Since the strain values of all sensors reach their starting values at the end of the pressure test, there is no danger of weakening the cryostat structure. Therefore the design and dimensions of the cryostat is verified as correct by the pressure tests. In sum, the maximum overpressure (of the safety valve) is set to 350mbar. The pressure tests yielded an elastic behaviour of the filled cryostat of up to 280mbar. The simulations show linear behaviour and no structural risk for an overpressure of up to 350mbar of a filled cryostat and the simulation and cryostat behave similarly up to 280mbar. Therefore the safety valve is the first failing component when it comes to excessive pressure on the cryostat and hence the cryostat structure is estimated to yield no risk of failing before 350mbar.

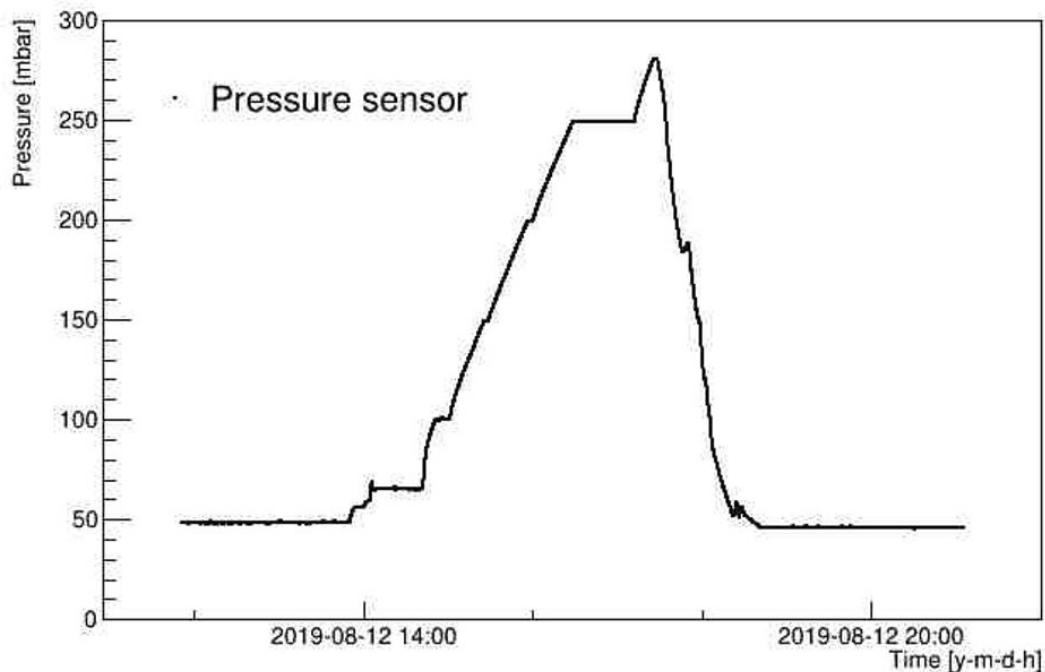


Figure 14: Pressure development at the 2nd NP02 pressure test on 12th August 2019.

⁵The safety valve is set to 350mbar overpressure.

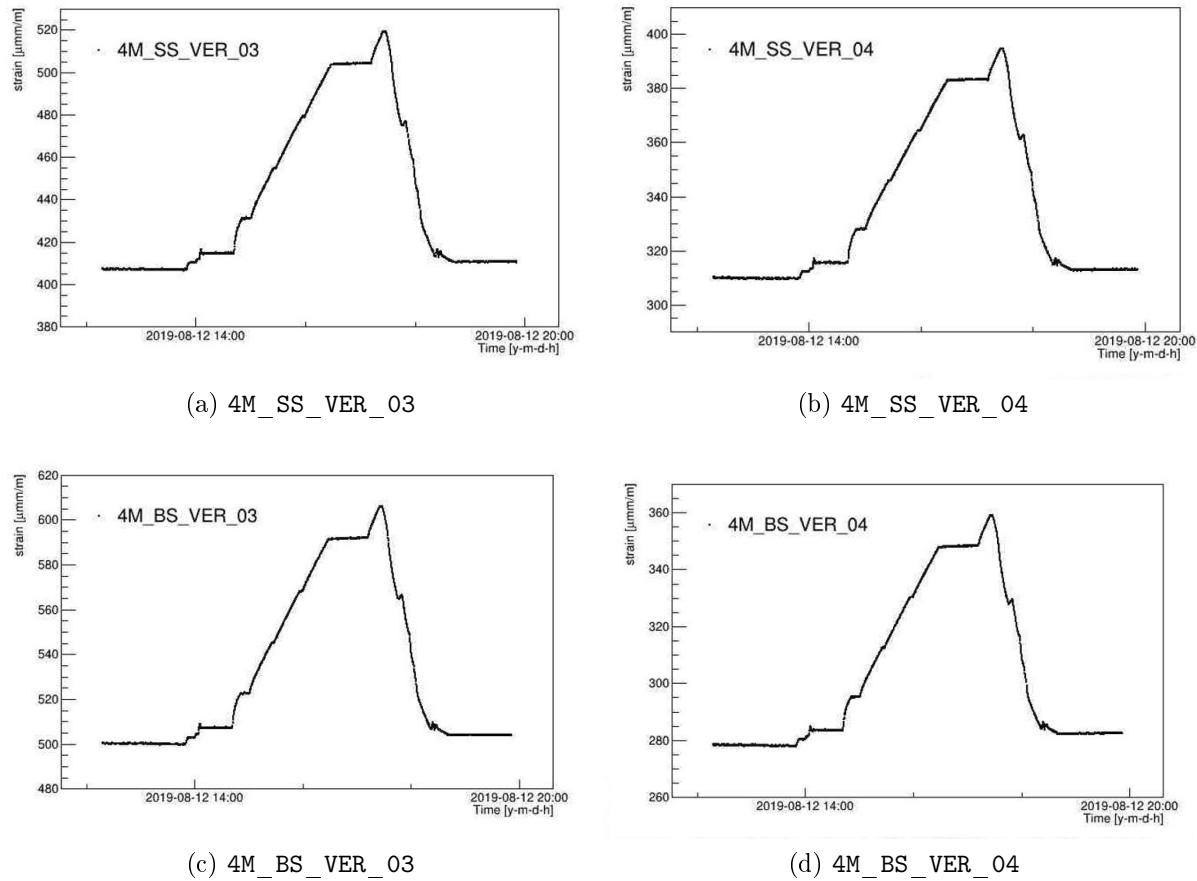


Figure 15: Strain development at the 2nd NP02 pressure test on 12th August 2019.

Moreover the success of the cryostat structure and its validation can be seen by observing the agreement between data and the simulation in fig. (16).

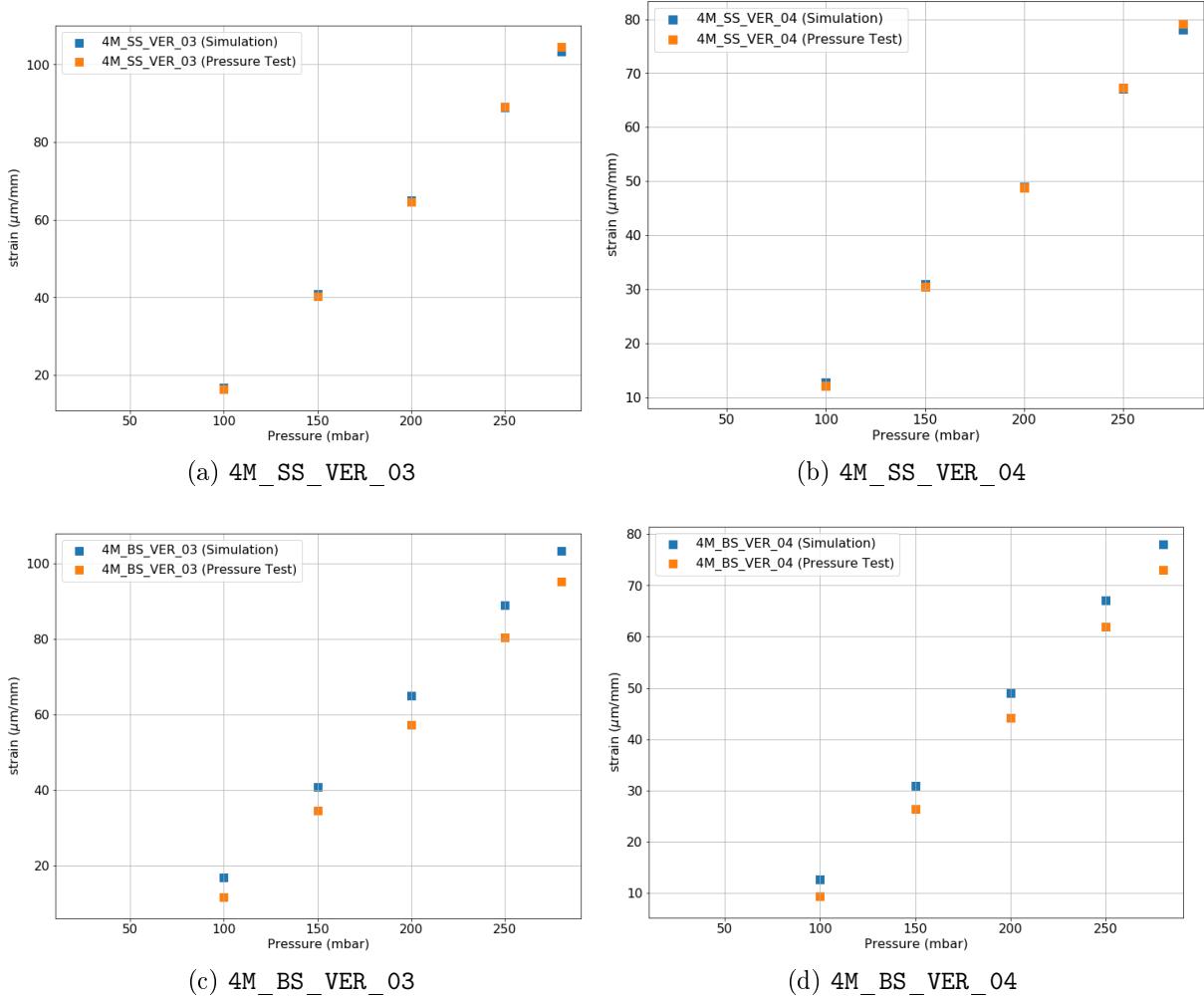


Figure 16: Strain development at the second NP02 pressure test on 12th August 2019.

8 Conclusion and Outlook

During this CERN Summer Student Project I worked on a strain gauge analysis with the purpose of validating the detector design and dimensions. This analysis commences the possibility of improving and fine-tuning cryostat deformation simulations. After commissioning NP02 and checking on the positive outcome of the pressure test, the cryostat confirms a strong and safe structure. Especially after the commissioning of NP04, new inputs from NP02 yield a robust and predictable structural behaviour of the ProtoDUNE cryostat. This will help to further estimate the structural behaviour of the future DUNE detector cryostat. The analysis gives hindsight on the possible structural behaviour of DUNE and more importantly helps to understand procedures of obtaining a reliable simulation model for the structure of DUNE. The strain gauge analysis confirms a safe ProtoDUNE detector design from the structural point of view, as envisioned by the simulations. The safety capabilities will be further enhanced by a stable and automatised monitoring of the data using the code that is shortly described in section (6.1) and described more in detail from page 21. In addition to the corrections applied in this analysis, the description of the code should allow one to modify, add or remove the corrections applied in this analysis to suit the purpose of new analyses. I hope the code written

and developed in my Summer Student Project at CERN serves as a framework of further studies and proves itself to be useful for future analyses of any kind and will be expanded to an inter-experimental basis. Data analysis of this kind is always going to be essential in order to succeed achieving the ambitious goals of the DUNE experiment.

Abbreviations

APA Anode Plane Assembly

CERN Conseil Européen pour la Recherche Nucléaire

CP Charge Parity

CPA Cathode Plane Assembly

CRP Charge Readout Planes

DP Dual-Phase

DUNE Deep Underground Neutrino Experiment

EHN1 Experimental Hall North 1

FD Far Detector

FEA Finite Element Analysis

FM Flash Matching

LArTPC Liquid Argon Time Projection Chamber

LBNF Long-Baseline Neutrino Facility

LEM Large Electrom Multiplier

PE Photo Electrons

PMT Photomultiplier Tube

SP Single-Phase

SPS Super Proton Synchrotron

SURF Sanford Underground Research Facility

References

- [1] The DUNE Collaboration, The DUNE Far Detector Interim Design Report, Volume 1: Physics, Technology & Strategies, arxiv:1807.10334 (2018).
- [2] N Charitonidis *et al* 2017, The beam lines design for the CERN neutrino platform in the CERN north area and an outlook on their expected performance, *J. Phys.: Conf. Ser.* 874 0012056.
- [3] The DUNE Collaboration, The Single-Phase ProtoDUNE Technical Design Report, arXiv:1706.07081v2 (2017).
- [4] R. Acciari, ProtoDUNE: prototyping the ultimate HEP neutrino detector, *Cavendish HEP Seminar Series*, Fermilab (2018).
- [5] DUNE: DUNE Deep Underground Neutrino Experiment, Neutrino Detectors, <https://www.dunescience.org/neutrino-detectors/> (2019).
- [6] ProtoDUNE-DP, NP02 Cryostat, <https://np02-slow-control.web.cern.ch/np02-slow-control/app/#!/cryostat> (2019).



Validation and study of the design and simulation model of the ProtoDUNE cryostats

Strain Gauge Analysis

CERN Summer Student Programme 2019

Technical details and documentation of the code

Lars Bathe-Peters⁶

Supervised by: Olga Beltramello, Andrea Zani , Alberto Rigamonti
(CERN)

20th September, 2019

Scope of this document

This documentation shall be used as a guideline in order to be able to run the code developed in this project. After data is extracted from a database and written into a .csv – file of the same format as it is used in this code, this code serves as a generic tool to generate plots for variables of choice and thus holds the potential to be extended and used for analyses beyond the scope of ProtoDUNE.

⁶mailto:lars_bathe-peters@fas.harvard.edu or lars.bathe-peters@cern.ch

9 Analysis code

The output values of the sensors installed on the cryostat of the ProtoDUNE Single-Phase (NP04) and Dual-Phase (NP02) detectors were plotted over time t . There are currently four plot modes implemented in this code, that can be adjusted according to any variables of choice. This code⁷ serves as a general tool to plot arbitrary variables with x - and y -values stored in a .csv – file. The current implemented plot modes

- strain
- displacement
- temperature
- pressure

can be extended to additional or different plot modes in `strain.C`. There are six scripts from the readout of the data until the generation of the plots:

- `RetrieveOracleDBdata.ipynb`
- `get_column_length.C`
- `create_variable_file.C`
- `make_variable_corrections.C`
- `merge_root_files.C`
- `make_plots.C`

Names of files and variables appearing in this script will be written like `these_example_words`.

10 Extract data from database (`RetrieveOracleDBdata.py`)

In order to extract data from the *NP02*- or *NP04*-slow control system, the python script `RetrieveOracleDBdata.ipynb` can be opened in 'jupyter notebook' and run with the following specifications:

10.1 Specify start and end date

The values will be stored starting after 12am of start date `start` and until 12am of the day prior to the end date `end`.

```
start=datetime(2019,7,10)
end=datetime(2019,7,16)
```

Figure 17: `start` and `end` in `RetrieveOracleDBdata.ipynb`.

⁷Github link: https://github.com/larsb-p/CERN_Project_Plottting.git

Besides the `start` and `end` nothing has to be changed. Once the data is extracted from the database into `.csv`-files of the format

```
data_files/sensors_dd_mm - dd_mm_yyyy.csv , e. g.  
data_files/sensors_08_07 - 09_07_2019.csv ,
```

the two days corresponding to the data in that `.csv`-file are specified in the name of the file. The first line of the `.csv`-files has to be blank and therefore manually inserted.

	A	B	C	D	E	F
1	Time47900797174811	TC_JS_02_A	Time47900830729243	TC_JS_02_B	Time47900847506459	BC_JS_03_A

	A	B	C	D	E	F
1	Time47900797174811	TC_JS_02_A	Time47900830729243	TC_JS_02_B	Time47900847506459	BC_JS_03_A
2	Time47900797174811	TC_JS_02_A	Time47900830729243	TC_JS_02_B	Time47900847506459	BC_JS_03_A

The names of the `.csv`-files containing the data that is to be plotted will be written manually into a `.txt`-file called `input_april_long.txt`-file:

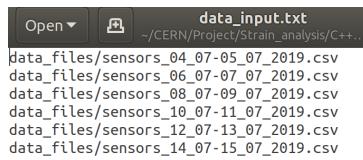


Figure 18: Example for `data_input.txt`.

11 Get number of values per sensor (`get_column_length.C`)

The number of values saved in the `.csv`-file is different for each sensor. If a value, e. g. strain, measured by a sensor changes more over a given time interval compared to another sensor, then the first sensor will save more values in the same time interval. Since data is readout more frequent for that sensor, the number of values saved in a given time interval is different for each sensor and will be saved in a `.root`-file called `column_count.root`. In here there is a TTree for each `.csv`-file specified in `input_april_long.txt`:

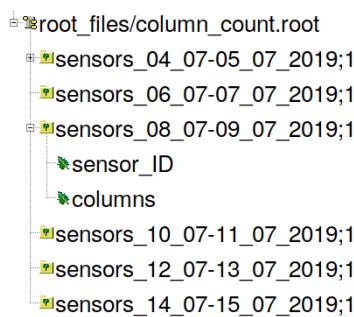


Figure 19: Example for `column_count.root`.

The TTree has two TBranches. In order to tag each sensor with a different label, the ID of the sensor is stored in `sensor_ID`. The number of values saved in the corresponding `.csv`-file is saved in `columns`. The number of entries in both TBranches corresponds to the number of sensors.

12 Save values to .root-file

(`create_variable_file.C(int x)`)

After `data_input.txt` and `column_count.root` are taken as an input for `create_variable_file.C(int x)`, the following values are saved in a `.root`-file called `root_files/file_* .root`. The `*` denotes the integer x that is used to loop over all sensors, so that the output is one `.root`-file for each sensor.

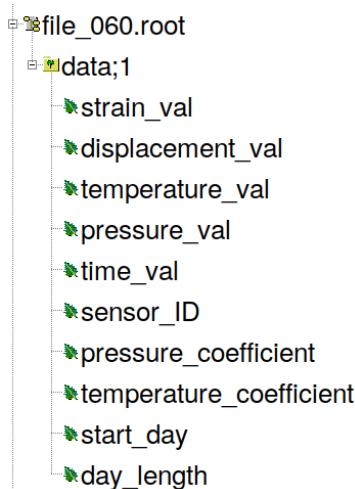


Figure 20: Example for `root_files/file_* .root`.

The path and file names for the sensors are then saved in `root_path_file.txt`. The path to the `root_files/file_* .root`-file must not contain spaces. The strain, displacement, temperature, pressure and time values are saved in the respective TBranches. Furthermore the sensor IDs are stored in it. The pressure and temperature coefficient value are the readout coefficients from `P_coeff_NP02.txt` and `T_coeff_NP02.txt` respectively. These `.txt`-files must not contain 0 (for example write $1e - 31$). `start_day` saves the day taken to determine the day length (If there is only one `.csv`-file in `data_input.txt`, `start_day` is the first day in that file. If there is more than one `.csv`-files in `data_input.txt`, `start_day` is taken from the second `.csv`-file.). Then `day_length` is the number of values starting from `start_day` until the the next day. In order for `day_length` to be determined, there has to be data for more than one day in the `.csv`-file.

13 Apply corrections to data

(`make_variable_corrections.C`)

`root_path_file.C` is the input for `make_variable_corrections.C` and adds a new TTree called `corrections` to the same TFile `root_files/file_* .root`-files.

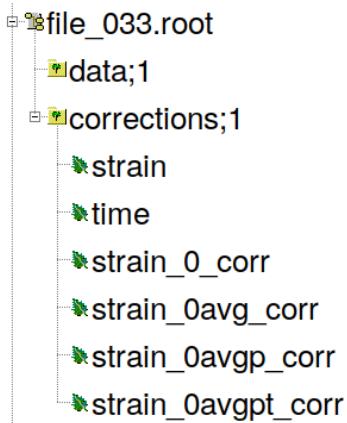


Figure 21: Example for `root_files/file_*.root` with the added TBranch called `corrections`.

In dependence on the `start_index` specified in `corrections`, the TBranches in the `corrections`-TTree contain the respective (reduced with respect to the number of entries in `data`-TTree) number of entries. These strain and time values are saved in `strain` and `time`. The dataset in `strain` is zeroed and saved in `strain_0_corr`. Then a moving average correction is applied and saved in `strain_0avg_corr`. The pressure correction is saved in `strain_0avgp_corr` and the temperature correction on top of that is stored in `strain_0avgpt_corr`.

14 Merge file_* .root-files (`merge_root_files.C`)

The `file_*.root`-files are then merged into one `.root`-file called `merged_file.root`. Hereby the TTrees `data` and `corrections` are added to a TChain and then written to `merged_file.root`.

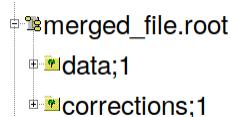


Figure 22: `merged_file.root` contains the merged TTrees `data` and `corrections`.

15 Generate plots (`make_plots.C`)

`merged_file.root` is taken as the input for `make_plots.C`. The output is a new `.root`-file called `plots.root`. For each sensor there is a `TGraph` saved in `plots.root`. If there is more fine tuning in the format of the plot needed, the `TCanvas` can be zoomed in, adjusted and modified in many ways manually from here. The plots are also saved in `.pdf`-format in the `plots/` directory.

```

root_files/plots.root
└─ DS_JS_01;1
└─ MC_BJS_02;1
└─ MC_BJS_03_A;1
└─ MC_BJS_03_B;1
└─ MC_LJS_02;1
└─ MC_LJS_03_A;1
└─ MC_LJS_03_B;1
└─ _3M_SS_VER_04;1
└─ _4M_SS_VER_04;1
└─ _3M_SS_VER_03;1
└─ _4M_SS_VER_03;1
└─ _4B_SS_HOR_01;1
└─ DS_SS_01;1
└─ _3B_SS_HOR_01;1
└─ _3B_SS_HOR_02;1
└─ _3B_SS_HOR_03;1
└─ _1MEM_SS_VER_01;1
└─ _1MEM_SS_HOR_02;1
└─ FRM_SS_VER_02;1
└─ _1MEM_SS_HOR_04;1
└─ _1MEM_SS_3AX_05_HOR;1
└─ _1MEM_SS_VER_03;1
└─ _3M_SS_VER_02;1
└─ FRM_SS_HOR_01;1
└─ FRM_SS_HOR_03;1
└─ _4M_SS_VER_02;1
└─ _2B_SS_HOR_01;1
└─ _3M_SS_VER_01;1
└─ _4M_SS_VER_01;1
└─ TT_contact_Bottom;1
└─ TT_ambience_Bottom;1
└─ MC_BSS_HOR_4B;1
└─ MC_BSS_HOR_3B;1
└─ MC_BSS_HOR_2B;1
└─ _4M_BS_VER_05;1
└─ _4M_BS_VER_04;1
└─ _4M_BS_VER_03;1
└─ _3B_BS_HOR_01;1
└─ _3B_BS_HOR_02;1
└─ _3B_BS_HOR_03;1
└─ _4M_BS_VER_02;1
└─ _2B_BS_HOR_01;1
└─ _4M_BS_VER_01;1
└─ TC_LS_02_A;1

```

Figure 23: `plots.root` containing chosen sensors of NP02.

The plots can then be displayed by double-clicking on the TGraph of the respective sensor.

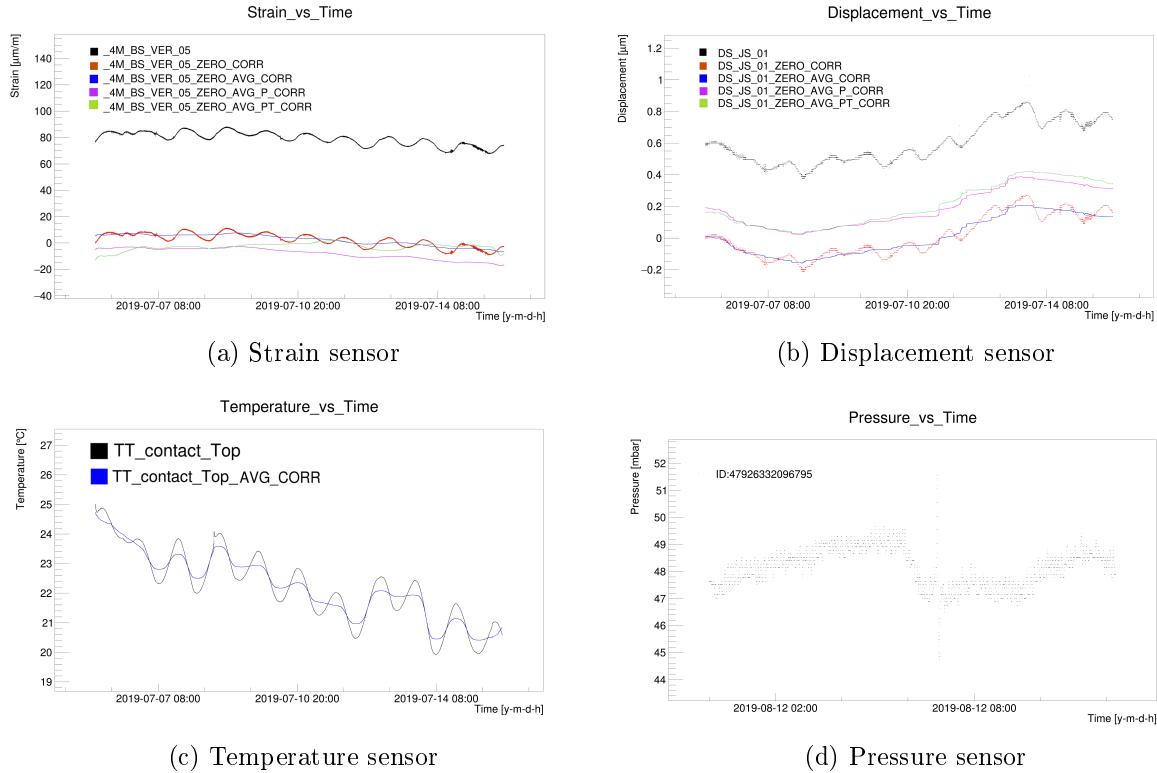


Figure 24: Example plots for sensors installed on the ProtoDUNE Dual-Phase detector cryostat.

16 Run all scripts (`run_it.sh`)

When running all scripts for the first time, type:

```
chmod -R +rwx root_files/
```

Then all scripts can be run automatically by typing

```
. run_it.sh
```

```
#!/bin/bash
rm 'txt_files/root_path_file.txt'
rm root_files/*_root
root -l -q -b 'get_column_length.C'
root -l -q -b 'create_variable_file.C(63)'
root -l -q -b 'make_variable_corrections.C'
rm 'root_files/file_TT_ambiance_Top.root'
mv root_files/column_count.root .
mv root_files/*.root root_files/file_TT_ambiance_Top.root
printf "File 'root_files/file_TT_ambiance_Top.root' with TTree 'corrections' has been created for the temperature correction.\n"
mv column_count.root root_files/
rm 'txt_files/root_path_file.txt'
for m in {0..63}; do
    root -l -q -b 'create_variable_file.C("$m")'
done
    root -l -q -b 'make_variable_corrections.C'
    root -l -q -b 'merge_root_files.C'
for m in {0..63}; do
    root -l -q -b 'make_plots.C("$m", "user_defined")'
done
```

Figure 25: `run_it.sh`-script.

In the loops of `run_it.sh` one can specify over which sensor to run the code (the indices in the loop correspond to the order of the sensors in the `.csv`-file). Since one sensor (now it is set to sensor `TT_ambiance_Top` (index 63)) is needed for the temperature correction of every strain and displacement sensor, there has to be a `file_TT_ambiance_Top.root`-file in the `root_files`/ directory. Therefore, if that file does not exist yet, `create_varaible_file.C(intx)` has to be run for that sensor before running the scripts on other sensors. This can be seen in the first few lines of fig. 25. If the temperature correction file already exists, one can uncomment all lines in `run_it.sh` until the for loops (`for min0..63; do`). or the index of that sensor has to be included in the loop of `run_it.sh`.

17 Future perspective of this work

I hope this documentation is enough of explanations to use this code for monitoring of the ProtoDUNE cryostat sensors. As the datasets plotted in this project are stored in a given `.csv`-format, this first step of extracting data like it is done in section (10) is the only step dependent on the database of an experiment and has to be implemented according to it when using this code. The scripts that follow after section (10) are generic scripts applicable to any `.csv`-file with that given format and can thus be used for plotting and automatic monitoring in any experiment or framework.

18 Appendix

The following shows plots of more chosen readout and corrected values of sensors intalled on the cryostat of NP02:

