Lehrstuhl für Mikro- und Nanosystemtechnik
TUM School of Computation, Information and Technology
Technische Universität München

Pressure Sensor Calibration

Wissenschaftliche Arbeit zur Erlangung des Grades Bachelor of Science (B.Sc.) an der TUM School of Computation, Information and Technology der Technischen Universität München.

Betreut von Heinrich, Ferdinand; Dorn, Christian
Prof. Dr-Ing. Hagelauer, Amelie

Eingereicht von Raed Tarar
Am Römerstein 39b
82205 Gilching
03771613

Begonnen am Gilching, den 01.05.2025

Eingereicht am München, den 02.08.2025

Abstract

This thesis was conducted in cooperation with Vectoflow GmbH. It addresses the development of an automated uncertainty analysis tool for pressure sensors, based on the DKD-R 6-1 guideline and ISO/IEC 17025. The goal was to extend an existing LabVIEW-based calibration routine by integrating a Python-based post-processing module, which is capable of calculating all uncertainty contributions in compliance with DKD-R 6-1 requirements. The tool processes raw calibration data and performs standardized uncertainty calculations for both analog and digital sensors. It then outputs data and plots intended for certificate generation. A modular software architecture was implemented, tested extensively with unit tests, and made callable from LabVIEW through a Python wrapper module. The resulting package replaces manual post-processing with an automated and traceable workflow. This work forms the foundation for DAkkS-accredited calibrations at Vectoflow and opens the possibility for future extensions, including fully automated certificate generation.

Kurzfassung

Diese Bachelorarbeit wurde in Zusammenarbeit mit der Vectoflow GmbH durchgeführt. Ziel war die Entwicklung eines automatisierten Tools zur Unsicherheitsanalyse von Drucksensoren, basierend auf der Richtlinie DKD-R 6-1 sowie der Norm ISO/IEC 17025. Hierzu wurde eine bestehende, auf LabVIEW basierende Kalibrierroutine um ein Python-Modul zur Nachverarbeitung erweitert, das sämtliche Unsicherheitsbeiträge normgerecht gemäß DKD-R 6-1 berechnet. Das Tool verarbeitet Rohdaten aus Kalibrierungen und führt standardisierte Unsicherheitsanalysen sowohl für analoge als auch digitale Sensoren durch. Die Ergebnisse werden anschließend in Form von Daten und Grafiken ausgegeben, die zur Zertifikatserstellung verwendet werden können. Die Software wurde modular aufgebaut, durch Unit-Tests abgesichert und über ein Python-Wrapper-Modul in LabVIEW integrierbar gemacht. Das daraus entstandene Softwarepaket ersetzt die manuelle Nachverarbeitung durch einen automatisierten und nachvollziehbaren Workflow. Diese Arbeit bildet die Grundlage für DAkkS-akkreditierte Kalibrierungen bei Vectoflow und eröffnet Perspektiven für zukünftige Erweiterungen, wie etwa die vollautomatische Zertifikatserstellung.

List of Abbreviations and Definitions

Abbreviation	Definition / Description
DKD-R 6-1	German calibration guideline for pressure sensors, specifying procedures and uncertainty evaluation methods. Published by PTB, used throughout this thesis.
DAkkS	Deutsche Akkreditierungsstelle — German national accreditation body for calibration and testing.
DUT	Device Under Test — The sensor or system being calibrated.
ISO/IEC 17025	International standard defining competence for testing and calibration laboratories.
Full Scale (FS)	Maximum measurement range of a sensor; used for relative uncertainty or error expressions.
DAQ	Data Acquisition System — Used for digitizing analog signals from sensors.
LabVIEW	Engineering software for automated data acquisition and control, used in the calibration setup.
Python	Programming language used for post-processing calibration data and uncertainty evaluation.
CSV	Comma-Separated Values — File format for recording and processing measurement data.
CPC6050	Precision pressure controller used as the reference standard in the calibration setup.
RSS	Root-Sum-Square — Method for combining standard uncertainties from independent sources.
SI Units	International System of Units — Standard measurement units, e.g., Pascal (Pa), Volt (V).
GUM	Guide to the Expression of Uncertainty in Measurement — International reference for uncertainty analysis.

List of Symbols

Symbol	Description
Δp	DUT reading deviation
p_{Display}	Displayed DUT reading
p_{avg}	Average DUT reading
δp_i	Pressure error term
S	Sensitivity (transfer coefficient)
U_{Output}	Output voltage
G	Gain factor
U_{Supply}	Supply voltage
p_{ref}	Reference pressure
$x_{m,j}$	DUT measurement at level j, series m
$x_{m,0}$	Zero reference value
$\bar{P}_{\text{up},j}$	Mean up-sweep DUT value
$\bar{P}_{\text{down},j}$	Mean down-sweep DUT value
$\bar{P}_{\text{mean},j}$	Overall mean DUT value
l	Number of up/down pairs
n	Number of measurement series
a	Half-width of uncertainty interval
$u(x)$	Standard uncertainty
r	Resolution
P	Measured pressure
f_0	Zero-point deviation

Kurzfassung

Symbol	Description
b'_{up}	Repeatability (up series)
b'_{down}	Repeatability (down series)
b'_{max}	Maximum repeatability
$h_{\text{mean},j}$	Mean hysteresis at pressure level j
$w(x)$	Relative uncertainty contribution
u_i	Combined uncertainty (digital)
U	Expanded uncertainty (digital)
w_i	Combined relative uncertainty (analog)
W	Expanded relative uncertainty (analog)
ΔS	Sensitivity deviation
$\bar{S}$	Mean sensitivity
S_j	Sensitivity at pressure level j
U_S	Uncertainty of sensitivity
U'	Maximum error span
K	Coverage factor

Contents

Abstract
Kurzfassung

List of Abbreviations and Definitions

List of Symbols

1. Introduction ... 1
 1.1. Problem Statement ... 1
 1.2. Motivation ... 3
 1.3. State of the Art ... 4
 1.4. Objectives ... 6
2. Main ... 7
 2.1. Theory ... 7
  2.1.1. Calibration and Setup ... 7
  2.1.2. ISO/IEC 17025 and DKD-R 6-1 Guidelines ... 8
  2.1.3. Sensors and Hardware ... 11
  2.1.4. Statistical Fundamentals for Sensor Calibration ... 12
 2.2. Methodology ... 14
  2.2.1. Software Tools ... 14
  2.2.2. Data Processing ... 14
  2.2.3. Uncertainty Analysis ... 15
  2.2.4. Calculations ... 17
  2.2.5. Plotting and Curvefit ... 20
 2.3. Implementation ... 22
  2.3.1. Input and Output variables ... 22
  2.3.2. Software Architecture ... 24
  2.3.3. Testing and Verification ... 28
 2.4. Results ... 30
  2.4.1. Automated Output and Workflow Integration ... 30
  2.4.2. Export files ... 31
3. Conclusion and Outlook ... 36
 3.1. Conclusion and Critical Reflection ... 36
  3.1.1. Conclusion ... 36
  3.1.2. Critical Reflection ... 36
 3.2. Outlook ... 36
Bibliography ... 39

Tools and Aids Used ... 42
Declaration ... 43
List of Figures ... 44
List of Tables ... 45

1. Introduction

Pressure measurement is critical for ensuring safety, performance, and regulatory compliance across both industrial and scientific applications, particularly in fields like aerospace, automotive, and process engineering. Accurate and reliable pressure data is essential for optimizing processes, verifying structural integrity, and meeting strict regulatory standards. Due to these demands, precise and repeatable calibration of pressure sensors is necessary to ensure traceable and trustworthy measurement results [1].

This thesis was conducted in collaboration with Vectoflow GmbH, which develops and manufactures high-precision, 3D-printed flow measurement probes. These devices are used in demanding aerodynamic environments where accurate and traceable pressure measurements are required, particularly for product validation, safety, and certification.

To ensure reliable measurements, pressure sensors must be calibrated regularly against known reference standards. Calibration not only enhances measurement accuracy but also supports compliance with internationally recognized quality systems such as ISO/IEC 17025 [2] and country-wide standards like the DKD-R 6-1.

1.1. Problem Statement

The current Pressure Sensor Calibration setup at Vectoflow, consisting of a Mensor CPC6050 pressure controller and a LabVIEW-based calibration routine, offers limited flexibility and does not fully comply with modern calibration standards. It supports only a subset of sensor types and lacks essential features needed for traceable and standards-compliant workflows. Specifically, it does not meet the documentation and uncertainty analysis requirements set out by DKD-R 6-1 and ISO/IEC 17025. Furthermore, the lack of automation introduces the potential for human error and reduces overall efficiency, an increasingly critical concern in high-precision environments.

A central limitation is the absence of uncertainty analysis in accordance with the DKD-R 6-1 guideline [3], which serves as the technical foundation for DAkkS-accredited pressure calibrations in Germany. Currently, measurement data must be manually exported and evaluated—a time-consuming, error-prone, and non-transparent process.

To address this, a Python-based post-processing tool will be developed. This tool will read raw calibration data, compute uncertainty contributions according to DKD-R 6-1, and calculate a complete uncertainty budget. It will be tightly integrated into the existing LabVIEW calibration routine, enabling users to perform the entire process with minimal manual effort.

Once the uncertainty analysis is in place, the workflow will be expanded to include sensor-specific calibration plots. These certificates will summarize all key results and ensure full traceability in accordance with ISO/IEC 17025. Additionally, potential hardware or procedural improvements may be investigated to further enhance measurement quality and system reliability.

1.2. Motivation

Pressure measurements are critical in many scientific and engineering domains. In high-stakes industries such as aerospace, the need for accuracy, reliability, and traceability continues to grow. It is no longer sufficient to achieve accuracy occasionally—pressure data must be consistently reliable and fully traceable.

Calibration is, therefore, indispensable. However, in addition to accuracy, other factors such as process efficiency, repeatability, and thorough documentation have become increasingly important. Many calibration setups still rely on manual steps, which reduces efficiency, introduces more opportunities for error, and limits scalability.

To overcome these challenges, many companies are now automating their calibration processes. Automation improves data integrity, reduces human involvement, and supports compliance with international standards.

This thesis builds on that trend. It aims to automate the existing calibration routine at Vectoflow, which currently lacks an integrated uncertainty analysis, plot generation, or automatic certificate creation. The goal is to integrate a Python-based post-processing system with the existing LabVIEW calibration environment and bring the process into full compliance with current norms.

1.3. State of the Art

Various methods exist for calibrating pressure sensors, each offering distinct advantages and limitations.

Deadweight Calibration is a primary method for generating pressure when known masses are applied to a piston-cylinder system. This technique is highly accurate and directly traceable to SI units due to its reliance on physical constants (mass, gravity, and area). However, it is relatively slow, sensitive to environmental conditions, and not easily automated [4].

Comparison Calibration is more practical for many industrial applications. In this method, both the sensor under test and a reference sensor are exposed to the same pressure, and their outputs are compared across a range of values. Digital devices such as the Mensor CPC6050 integrate both pressure control and high-precision reference measurement capabilities, making this approach more suitable for automation [4].

No matter the calibration method, other critical factors must be taken into account, such as **Temperature Compensation**. Pressure sensors often exhibit measurement drift due to temperature variations. To correct this, many calibration systems implement temperature-dependent correction models through hardware or software [5, p. 2].

Dwell time is also a critical factor in this analysis. Subsequent to a pressure change, sensors must undergo a period of stabilization before accurate readings can be obtained. Insufficient dwell time has been demonstrated to result in errors caused by hysteresis or signal lag [6].

Lastly, **automated uncertainty analysis** has become increasingly standard. Modern calibration systems incorporate software that automatically calculates measurement uncertainties, ensuring consistency, reducing human error, and saving time [7].

There are many standards and guidelines for pressure sensor calibration. Some of the most prominent include the German "Deutscher Kalibrierdienst" (DKD) and the "National Institute of Standards and Technology" (NIST) from the United States. While both standards aim for reliability and traceability, their methodologies differ. The DKD-R 6-1 offers a structured form and formal methods, which include predefined calibration sequences, minimum dwell time, environmental correction, and a sensor-type-specific uncertainty budget [3, p. 8–26]. In contrast, the NIST allows for more flexibility and permits laboratories to adapt procedures as long as they comply with the "Guide to the Expression of Uncertainty in Measurement" (GUM) principles [8, p. 14–18]. GUM provides a standardized method for combining statistical and non-statistical uncertainty components [9]. Even though NIST offers more procedural versatility, this thesis adopts the DKD-R 6-1 as the normative basis due to its accuracy and traceability.

While this thesis is based on the structured framework defined in DKD-R 6-1, current research shows that alternative approaches to uncertainty evaluation are gaining importance. In particular, beyond traditional methods such as the root-sum-square (RSS) model applied by DKD-R 6-1, recent research highlights the increasing relevance of **Monte Carlo methods (MCM)**. These stochastic approaches estimate uncertainty by repeatedly simulating the measurement model with random input values. Smith et al. [10] have demonstrated how Monte Carlo simulations can be integrated into Digital Calibration Certificates (DCCs), thereby improving digital traceability, machine readability, and the transparency of uncertainty reporting.

A further development in this context is the use of **virtual experiments (VEs)** as proposed by Hughes et al. [11]. This JCGM 101-compliant method uses computational models to simulate the whole measurement process, including its associated uncertainties. Unlike strictly analytical methods, virtual experiments allow for more flexibility and better representation of complex system behavior. These modern approaches do not replace classical frameworks such as DKD-R 6-1 but rather extend them and offer new possibilities for uncertainty modeling, particularly in automated or simulation-based environments.

1.4. Objectives

The primary objectives of this thesis are to:

- Develop a Python-based post-processing tool to automate calibration data analysis and measurement uncertainty calculation.
- Generate plots for visualization and data analysis
- Fully integrate this tool into the existing LabVIEW-based calibration workflow.
- Prepare data for generating calibration certificates
- Implement curve fit of the calibration and computation of sensor calibration coefficients
- Improve the overall calibration process efficiency, accuracy, and traceability.
- Ensure full compliance with relevant calibration standards, including ISO/IEC 17025 and DKD-R 6-1.

This approach is intended to enhance the calibration capabilities at Vectoflow, widen support for various sensor types, and improve the reliability and transparency of the measurement process.

2. Main

2.1. Theory

2.1.1. Calibration and Setup

Calibration is defined as the comparison of the pressure measurement of a device under test (DUT) to a reference standard. In the scope of the thesis, the DUT will always be a pressure sensor. The calibration procedures must adhere to national and international standards to ensure measuring instruments remain within specified tolerances of their manufacturer's accuracy [1].

Figure 1 presents a simplified sketch of the existing setup and the contributors to the measurement uncertainty. The calibration setup consists of the sensors, which are connected to the MENSOR CPC 6050 with a tube system and are then pressurized by the reference device. Then, the measurements are recorded using the same reference device and provided to the LabVIEW workflow.

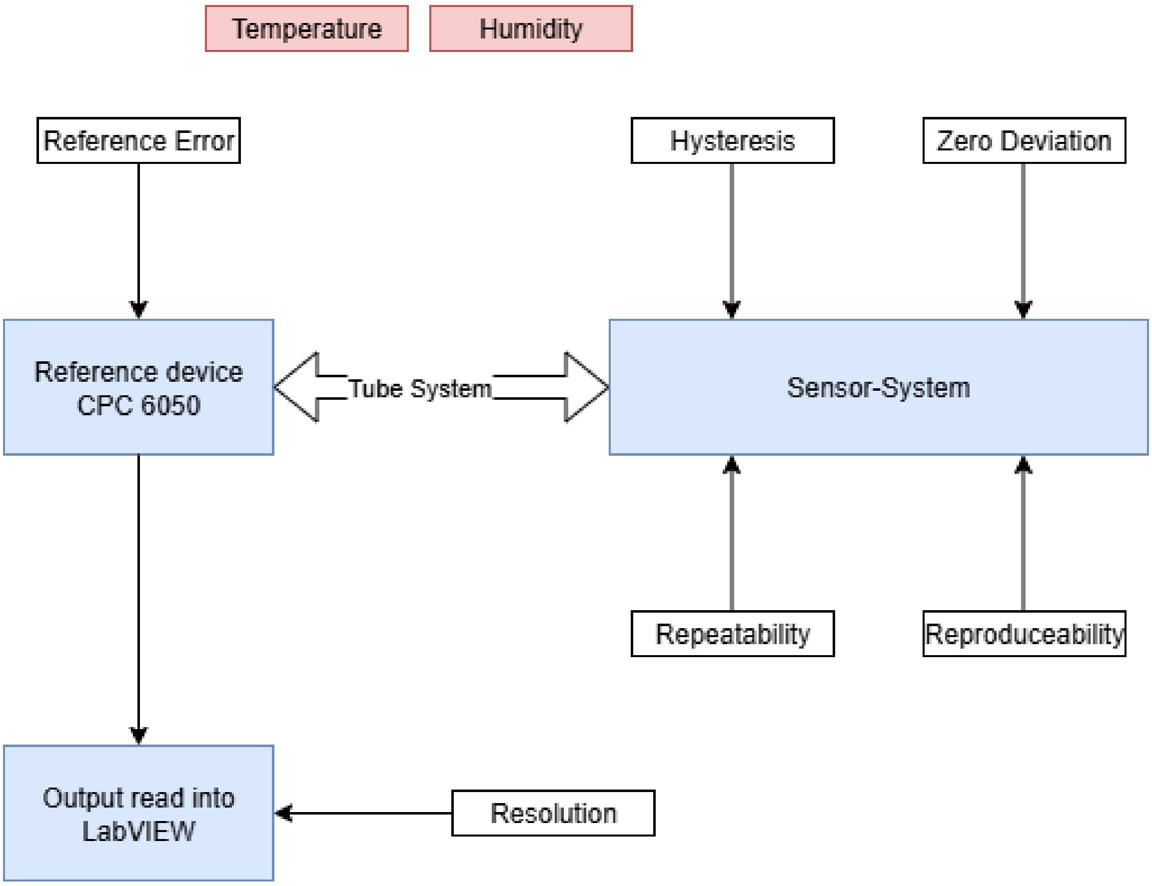

Figure 1 Calibration Setup with respective measurement uncertainty contributors. Marked in red are environmental conditions, which do not contribute to the measurement uncertainty but need to be stable for a successful calibration.

2.1.2. ISO/IEC 17025 and DKD-R 6-1 Guidelines

As the introduction mentions, some guidelines and requirements must be followed during calibration to be considered traceable and reliable. One of those is the DKD-R 6-1 (Deutscher Kalibrierdienst), which is used as a guideline for DAkkS accreditation. Furthermore, for the aerospace industry, the calibration process must adhere to the ISO/IEC 17025 norm. Which also has its own requirements for traceability and accuracy. In the following section, the two guidelines shall be explained.

DKD-R 6-1: Within the scope of this thesis, the DKD-R 6-1 (2016) will be used as a guideline, because the total calibration will be DAkkS accredited, for which the DKD-R 6-1 is used as a guideline. One of the main motivations for this was that the wind tunnels at Vectoflow are already DAkkS-accredited.

Furthermore, the DKD-R 6-1 requires documenting the environmental conditions, i.e., temperature during calibration, outside air pressure, and non-condensing humidity (under 100%). The conditions needs to be recorded, but do not affect the calibration if stable. However, the documentation of environmental conditions helps to assert whether a failed calibration was due to unstable environmental conditions or not.

Different calibration sequences are also outlined in the following table 3.

Criteria	Seq. A	Seq. B	Seq. C
Target uncertainty (% span)	< 0.1	0.1–0.6	> 0.6
Min. number of measurement points	9	9	5
Number of preloadings	3	2	1
Load change + wait time [s]	> 30	> 30	> 30
Wait time at upper limit [min]	2	2	2
Measurement series (up)	2	2	1
Measurement series (down)	2	1	1

Table 3 Calibration Sequences according to DKD-R 6-1 p. 12 table 1 [3]

The different Sequence names refer to different measurement uncertainty levels, where Sequence A is the most accurate and Sequence C the least accurate. Due to the various criteria that need to be met for each Sequence class. In the scope of this thesis, Sequence A will be used to get the most accurate calibration results.

Figure 2 illustrates the full calibration sequence structure, including preloadings, upward and downward series, and the required dwell times.

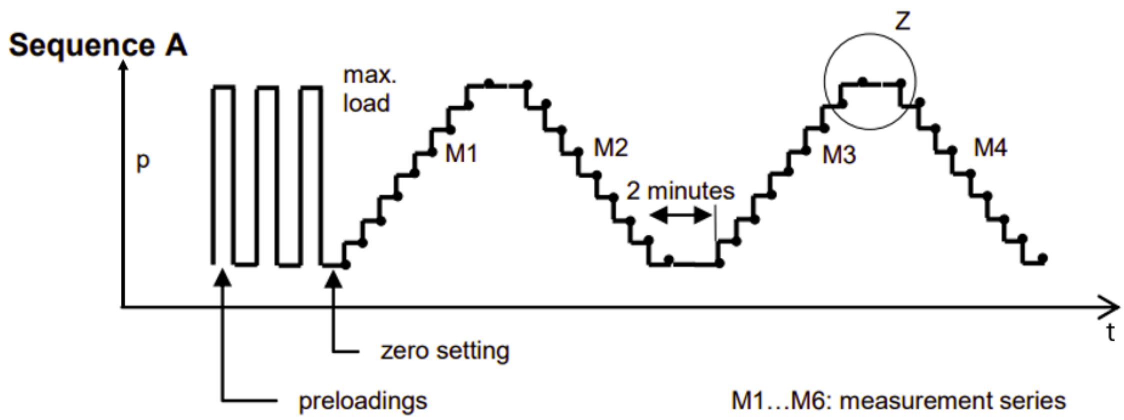

Figure 2 Visualization of the calibration sequence A (p. 13 [3])

The number of measurement points has to be equally distributed. As shown in Figure 2, the pressure increases from the lowest pressure point to the highest during the upward measurement series, and then decreases from the highest to the lowest pressure value during the downward measurement series.

Preloadings refer to full-scale pressure cycles applied before the actual calibration to stabilize sensor response. In the figure 2, these are visible at the beginning of the sequence. Before starting the actual calibration, the DUT has to go through between 1 and 3 preloadings, depending on the sequence class. This ensures the repeatability and stability of the measurements. Following the preloading, the reference reading is set to zero.

To ensure each measurement is stable and reliable, a waiting time of at least 30 seconds must be observed after each pressure change. This allows the sensor to settle before a value is recorded. In addition, when the maximum pressure is reached, the pressure must be held for at least 2 minutes before continuing with the downward series. This helps to account for pressure creep and ensures more consistent results (p. 12 [3]).

ISO/IEC 17025:
A fundamental requirement in calibration and metrology is compliance with ISO/IEC 17025, the international standard for the competence of testing and calibration laboratories. This standard provides a general framework to ensure the validity and traceability of measurement results.

Key principles of ISO 17025 include:

- Documenting and quantifying measurement uncertainty.
- Ensuring traceability to national or international standards.
- Maintaining environmental control during measurements.
- Using traceable instruments.

In the context of this thesis, several measures were taken to align with ISO 17025:

- All input quantities (e.g., pressure, voltage) are provided exclusively in SI units (e.g. Pascal, Volt), ensuring traceability.
- The calibration setup relies on DAkkS-accredited equipment (e.g., pressure controller CPC 6050).
- Although environmental parameters such as temperature and humidity are not logged dynamically in this setup, their stability was ensured during all calibration procedures.

These steps help meet the traceability, uncertainty evaluation, and documentation requirements outlined in ISO/IEC 17025 [2].

2.1.3. Sensors and Hardware

The calibration setup used in this study is centered around the **MENSOR CPC6050 Precision Pressure Controller**, a high-accuracy instrument designed for calibrating low- and medium-range pressure transducers. The CPC6050 allows for precise pressure generation and regulation, offering an accuracy up to $\pm 0.008\%$ Full Scale. It is used as the reference pressure generator throughout all calibration procedures.

Pressure Sensors

The calibrated sensor is called DUT (Device under Test). Two different types of pressure sensors were considered for use: absolute and differential.

Absolute pressure sensors measure pressure in relation to a vacuum, while differential pressure sensors measure the difference in pressure between two chambers. This difference makes differential pressure sensors unaffected by changes in atmospheric pressure [12]. The created workflow is able to work with both sensors.

Two different types of pressure sensors will be used in the calibration setup. The main difference lies in how the signal is put out. This also changes how the uncertainties are calculated.

Analog sensors output a voltage, e.g. between 0 and 5 Volts, corresponding to a specific reference pressure. For analog sensors, the uncertainty contributions are calculated using the product/quotient model (see equation 2.10), which uses relative uncertainty contributions(W). Additionally, the data from an analog sensor must be read using a Data Acquisition System (DAQ).

Digital sensors, on the other hand, output directly in a pressure unit, eg. Pascal. Since the reference and DUT have the same unit, measurement uncertainty can then be calculated using the absolute relative uncertainty contributions (U).

	Analog Sensor	**Digital Sensor**
Output Signal	Signal is Voltage corresponding to pressure level	Signal is Pressure
Uncertainty Calculation model	Product/Quotient Model	Sum/Difference Model
Uncertainty display	Relative (W)	Absolute (U)
DAQ required	Yes	No

Table 4 Analog and Digital Sensor comparison

2.1.4. Statistical Fundamentals for Sensor Calibration

A clear understanding of key statistical concepts is essential for evaluating measurement data and uncertainty in pressure sensor calibration. This section introduces the basic statistical background used in the thesis.

1. Measurement Deviation

The measurement deviation quantifies how much the output of a sensor (Device Under Test, or DUT) differs from a known reference (see DKD-R 6-1, p.19, eq.(8) [3]):

$$\Delta P = P_{\mathrm{DUT}} - P_{\mathrm{REF}} \tag{2.1}$$

2. Confidence Intervals and Coverage Factor k.

To express how certain a result is, the measurement uncertainty U is multiplied by a coverage factor **k** to get the expanded measurement uncertainty.

A common choice is $k = 2$, which corresponds to a 95% confidence level — meaning the true value is expected to lie within $\pm U$ of the measured result in 95 out of 100 cases (see DKD-R 6-1, p. 14 [3]).

3. Sensitivity.

Sensitivity is the ratio between signal change and pressure change. It describes how strongly the sensor responds to pressure:

$$S = \frac{\Delta\text{Signal}}{\Delta\text{Pressure}} \tag{2.2}$$

In this thesis, sensitivity is obtained via linear regression between reference pressures and measured signals. The slope of this regression line represents the sensor's sensitivity (derived from DKD-R 6-1, p. 25 [3]).

4. Linear Regression.

Linear regression is a statistical method used to model the relationship between two variables by fitting a straight line to observed data. In the context of sensor calibration, it quantifies how the output signal changes in response to variations in pressure.

The regression line is typically expressed as [13]:

$$Y = aX + b \tag{2.3}$$

where Y is the measured signal, X is the applied reference pressure, a is the slope (sensitivity), and b is the offset (deviation).

The slope a indicates how much the signal changes per unit of pressure and is therefore interpreted as the **sensitivity** of the sensor. A higher slope corresponds to a more responsive sensor. In this thesis, the sensitivity is determined by applying linear regression to the measured DUT output versus the known reference pressure values.

5. Type A and Type B Uncertainty.

- **Type A** uncertainty comes from statistical calculations (e.g., standard deviation).
- **Type B** uncertainty is based on manufacturer specs, calibration certificates, or assumptions (e.g., resolution, zero offset, hysteresis).

(see DKD-R 6-1, p. 16 [3])

6. Combined Uncertainty.
All relevant uncertainty sources — Type A and B — are combined using the root-sum-square (RSS) method (derived from DKD-R 6-1, p. 14 [3]):

For digital sensors, the absolute uncertainty is calculated:

$$u_c = \sqrt{u_1^2 + u_2^2 + \cdots + u_n^2} \tag{2.4}$$

This yields the combined measurement uncertainty u_c, which is then expanded using the coverage factor and is calculated in pressure units:

$$U = k \cdot u_c \tag{2.5}$$

For analog sensors, the relative uncertainty is calculated:

$$w_c = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2} \tag{2.6}$$

This yields the combined relative measurement uncertainty w_c, which is then expanded using the coverage factor and is calculated in percent:

$$W = k \cdot w_c \tag{2.7}$$

2.2. Methodology

The following section will describe the methods used for the thesis.

2.2.1. Software Tools

Data acquisition is performed via LabVIEW, which directly interfaces with the DAQ system and pressure controller. Data analysis and uncertainty evaluation are executed in Python using custom scripts by DKD-R 6-1 guidelines.

LabVIEW:
The NI LabVIEW 2021 32-bit version is used as a program to record the calibration data from the MENSOR CPC 6050. LabVIEW is a graphic programming environment that uses visual blocks connected by wires and is widely used in engineering applications.

Python:
Python is used for reading data, uncertainty analysis, and plotting. The decision to use Python in post-processing stems from the need for automation, which is easier to implement in Python. Moreover, Python offers greater user-friendliness than LabVIEW, as it is more widely known at Vectoflow.

External Python libraries, such as **numpy** 1.26[14], **pandas** 1.5.3[15], **matplotlib** 3.7.1[16], and the internal company library **vector-read-write**, were used. These older versions were chosen to ensure compatibility with Python 3.9 (32-bit) and the **vector-read-write** module.

The Python and library versions were explicitly selected for compatibility with the existing LabVIEW setup. Additionally, the Python script was executed in a virtual environment to manage these library dependencies and to isolate the Python tool from the system environment.

IDE and GIT:
The integrated development environment (IDE) used for most software development was Visual Studio Code, which supported efficient code editing and debugging. Version control was managed with Git to track changes and roll back to previous working versions.

2.2.2. Data Processing

The target for the data processing was defined as using the Calibration Data for an automatic uncertainty calculation in Python. The data itself was given as a CSV file. The dataset included units, values, and data types that were verified to be valid. Additionally, the number of measurement series was confirmed to be four, consistent with Sequence A from the DKD-R 6-1 guideline.

There were multiple solutions at hand to get the data. One used a Pandas Dataframe to read from the file, and the other used an internal tool in the company to get the data. Furthermore, how to process the data had to be decided. Divide it into a measurement series first, and then process it or read the sensor data at once.

The solution chosen at the end was to read the calibration data with the internal tool and divide the Calibration data into sensors (Device under Test) and then further into measurement series. The raw calibration data is given via I2C/SPI for a digital sensor in Pascal or as a voltage for analog sensors. Both values are read into LabVIEW. Multiple data points are measured per pressure level, and then those different measurement series are exported into a comma-separated value file and saved locally. Then the CSV file is read into the Python routine using an internal library. After that, the data is processed in the uncertainty analysis Python package, and then the processed data is exported.

2.2.3. Uncertainty Analysis

The task was to read in calibration data from the calibration setup and automatically perform the uncertainty analysis, which includes uncertainty contributors and other related values, to be calculated automatically and exported as a comma-separated values (CSV) file. The package should be in Python and callable with a VI (function) in LabVIEW. So the user only has to input the data for the calibration and where the user wants the data exported.

Measurement uncertainties were evaluated in accordance with the procedure outlined in the DKD-R 6-1 (p. 26 [3]).

The following table 5 summarizes the contributing factors to the measurement uncertainty.

Uncertainty Contributor	**Type**	**Explanation and Relevance**
Reference Standard Uncertainty	B	From the DAkkS certificate of the reference device (e.g., CPC6050). Represents the traceable uncertainty of the calibration device. In this application, given as 0.01% FS.
Resolution	B	Smallest detectable change in DUT output. Determined by output precision: In the existing routine 10^{-6} = 0.000001.
Repeatability	A	Standard deviation of repeated measurements under identical conditions. Reflects short-term noise.
Reproducibility	A/B	Variations across different setup environments (e.g. location or mount changes). Reproducibility is irrelevant for the current setup, because the sensor is not calibrated with multiple mounts.
Zero Offset	B	Deviation at zero pressure due to aging, initial error, or drift. Significant in absolute or low-pressure sensors.
Hysteresis	B	Difference in sensor output when approaching pressure from increasing (upwards series) vs. decreasing (downwards series).

Table 5 Uncertainty Contributors in Pressure Sensor Calibration

2.2.4. Calculations

In the following section, the calculations that were later implemented in Python shall be described and explained.

Sum/Difference Model (e.g., pressure deviation)

The Sum/Difference Model applies to measurements of sensors with the same measurement unit and reference unit. This is specifically the case for a digital sensor (see DKD-R 6-1, p.15, eq.(2) [3]):

$$\Delta p = p_{\text{Display}} - p_{\text{avg}} + \sum_{i=1}^{N} \delta p_i \tag{2.8}$$

This simplifies to the following equation, as no correction term is applied:

$$\Delta p = p_{\text{Display}} - p_{\text{avg}} \tag{2.9}$$

Product/Quotient Model (e.g., transfer coefficient)

The product model is used for analog sensors. The transfer coefficient S is calculated by dividing the measured output voltage by the product of the gain G, supply voltage U_{Supply}, and the applied reference pressure (derived from DKD-R 6-1, p.25, eq.(16) [3]):

$$S = \frac{U_{\text{Output}}}{G \cdot U_{\text{Supply}} \cdot p_{\text{ref}}} \tag{2.10}$$

This simplifies to the following form because the measured output already includes G and U_{Supply}:

$$S = \frac{U_{\text{Output}}}{p_{\text{ref}}} \tag{2.11}$$

Mean DUT value

$x_{m,j}$ represents the DUT measurement at pressure level j for measurement series m, and $x_{m,0}$ is the reference value from the zero-point (which needs to be subtracted to get the corrected mean value).

Then, the mean values are computed as follows (see DKD-R 6-1, p.32, eq.(28) [3]):

$$\bar{P}_{\text{up},j} = \frac{1}{l} \sum_{m=1,3} (x_{m,j} - x_{m,0}) \tag{2.12}$$

$$\bar{P}_{\text{down},j} = \frac{1}{l} \sum_{m=2,4} (x_{m,j} - x_{m-1,0}) \tag{2.13}$$

$$\bar{P}_{\text{mean},j} = \frac{\bar{P}_{\text{up},j} + \bar{P}_{\text{down},j}}{2} \tag{2.14}$$

Here:

- $l = \frac{n}{2}$ is the number of up/down measurement pairs,
- n is the total number of measurement series.

Uncertainty Rectangular Distribution
This formula calculates the standard uncertainty $u(x)$ assuming a rectangular (uniform) distribution, where all values within the interval $[x_-, x_+]$ are equally likely (see DKD-R 6-1, p.17, Table 2 [3]):

$$u(x) = \frac{a}{\sqrt{3}} \quad \text{where } a = \frac{x_+ - x_-}{2} \tag{2.15}$$

Uncertainty Normal Distribution
This equation calculates the standard uncertainty when the underlying distribution is assumed to be normal, and a is the half-width of the confidence interval (see DKD-R 6-1, p.17, Table 2 [3]):

$$u(x) = \frac{a}{2} \tag{2.16}$$

Resolution
The resolution r is defined as the smallest change in a measurement that can be reliably displayed. It is given as 10^{-6} based on the raw calibration data. However, a case distinction must be made between analog and digital sensors.

For digital sensors, the uncertainty contribution is calculated by dividing the resolution by the square root of 3, assuming a rectangular distribution (see DKD-R 6-1, p.17, Table 2 [3]):

$$u(r) = \frac{r}{\sqrt{3}} \tag{2.17}$$

In the case of analog sensors, this uncertainty must be expressed relative to the measured pressure value of the device under test (DUT). The resulting relative uncertainty contribution is given by (see DKD-R 6-1, p.27 [3]):

$$w(r) = \frac{r}{\sqrt{3} \cdot P} \tag{2.18}$$

Zero Point Deviation
The zero deviation f_0 is defined as the maximum absolute difference between the up and down series at the zero reference point, across all measurement cycles (see DKD-R 6-1, p.29, eq.(24) [3]):

$$f_0 = \max_i \left(\left| x_{i,0}^{\text{down}} - x_{i,0}^{\text{up}} \right| \right) \tag{2.19}$$

Repeatability
Repeatability b' is calculated from two complete measurement cycles. It is defined as the

maximum absolute difference between the corrected second and first measurement cycle for both up and down series (see DKD-R 6-1, p.28, eq.(25) [3]):

$$b'_{\text{up}} = |(x_{\text{M3}} - x_{\text{M3},0}) - (x_{\text{M1}} - x_{\text{M1},0})| \tag{2.20}$$

$$b'_{\text{down}} = |(x_{\text{M4}} - x_{\text{M3},0}) - (x_{\text{M2}} - x_{\text{M1},0})| \tag{2.21}$$

$$b'_{\text{max}} = \max\left(b'_{\text{up}}, b'_{\text{down}}\right) \tag{2.22}$$

Hysteresis

The hysteresis at each pressure point is the mean of the absolute differences between the corrected up and down values over all cycles (see DKD-R 6-1, p.28, eq.(27) [3]):

$$h_{\text{mean},j} = \frac{1}{n} \sum_{i=1}^{n} \left| \left(x_{i,j}^{\text{down}} - x_{i,j}^{\text{up}} \right) \right| \tag{2.23}$$

Reproducibility

Reproducibility refers to variation between different mountings or repeated calibrations of the same sensor. In this setup, reproducibility is not considered, since each DUT is calibrated only once without repositioning.

Uncertainty Contributors

The values f_0, h, and b'_{max} are then further calculated into the uncertainty contributors using coverage factor $k = 2$ and the square root of 3 as the dividing factor, resulting in the following equation for digital sensors (see DKD-R 6-1, p.27 [3]):

$$u(x) = \sqrt{\frac{1}{3} \cdot \left(\frac{x}{2}\right)^2} \tag{2.24}$$

For analog sensors, the value is just divided by the measured pressure value of the DUT (see DKD-R 6-1, p.27 [3]):

$$w(x) = \frac{1}{P} \cdot \sqrt{\frac{1}{3} \cdot \left(\frac{x}{2}\right)^2} \tag{2.25}$$

Absolute Measurement Uncertainty

The absolute measurement uncertainty for digital sensors is calculated by combining all uncertainty contributors:

$$u_i = \sqrt{u_r^2 + u_{\text{ref}}^2 + u(f_0)^2 + u(b')^2 + u(h)^2} \tag{2.26}$$

Absolute Expanded Measurement Uncertainty

$$U = 2 \cdot u_i \tag{2.27}$$

Relative Measurement Uncertainty
For analog sensors, the relative measurement uncertainty is computed instead:

$$w_i = \sqrt{w_r^2 + w_{\text{ref}}^2 + w(f_0)^2 + w(b')^2 + w(h)^2} \tag{2.28}$$

Relative Expanded Measurement Uncertainty

$$W = 2 \cdot w_i \tag{2.29}$$

Delta Sensitivity
Deviation between one-point sensitivity S' and the sensitivity measured at each pressure level (derived from DKD-R 6-1, p.47, Table D3 [3]):

$$\Delta S = S' - S_j \tag{2.30}$$

Single-Figure Sensitivity
Average of all sensitivities, where S_j is the sensitivity at one pressure level calculated as DUT_value / Ref_value (derived from DKD-R 6-1, p.47, Table D3 [3]):

$$\bar{S} = \frac{1}{n} \sum_{j=1}^{n} S_j \tag{2.31}$$

Uncertainty of Sensitivity $U(S_j)$
The uncertainty of the sensitivity is calculated by multiplying the relative expanded measurement uncertainty with the sensitivity per pressure level (derived from DKD-R 6-1, p.47, Table D3 [3]):

$$U(S_j) = W \cdot |S_j| \tag{2.32}$$

Maximum Sensitivity Error Span U'
The maximum error span is then calculated by adding $U(S_j)$ and the deviation ΔS (derived from DKD-R 6-1, p.47, Table D3 [3]):

$$U' = |\Delta S| + U(S_j) \tag{2.33}$$

2.2.5. Plotting and Curvefit

Curvefit

A linear curve fit was chosen to extract correlation coefficients and standard deviations. This method was selected because it directly shows whether the calibration behaves linearly across different reference pressure levels. Ideally, the correlation coefficient R is 1. As the value decreases, linearity worsens, indicating poorer sensor performance. This makes the linear fit a simple and effective way to assess calibration quality.

Plots

To support DKD-R 6-1-compliant calibration evaluation and internal quality assessment, selected plots are generated within the uncertainty analysis package. These graphics complement the numerical uncertainty analysis and make the calibration validation easier.

- **Linear Regression DUT/REF:** Used to assess the linearity of the DUT response across the pressure range. A linear curve fit provides insight into slope (sensitivity), offset, and correlation. This enables both visual and numerical assessment of non-linear behaviour in the sensors. This plot enables an easy check for the success of the calibration.
- **Measurement deviation with expanded measurement Uncertainty:** Displays calculated deviation values and expanded measurement uncertainty for digital sensors. For analog sensors, the sensitivity deviation and the uncertainty of the sensitivity are plotted. This is essential to evaluate compliance with sensitivity and deviation tolerances defined by DKD-R 6-1 guidance.
- **Measurement Deviation ΔP with Min–Max Scatter:** Used to visualize individual deviation series and highlight pressure point scatter. This helps identify repeatability issues and assess conformance to full-scale deviation limits. This plot is required for internal quality assessment.

2.3. Implementation

In the following section, the implementation shall be described with an emphasis on the software and the testing thereof.

2.3.1. Input and Output variables

The flowchart figure 3 displays the inputs and outputs of the post-processing tool. The Python uncertainty analysis package is connected to a LabVIEW interface with a wrapper file in Python. Which, when called by LabVIEW, executes the uncertainty analysis with the displayed inputs and outputs.

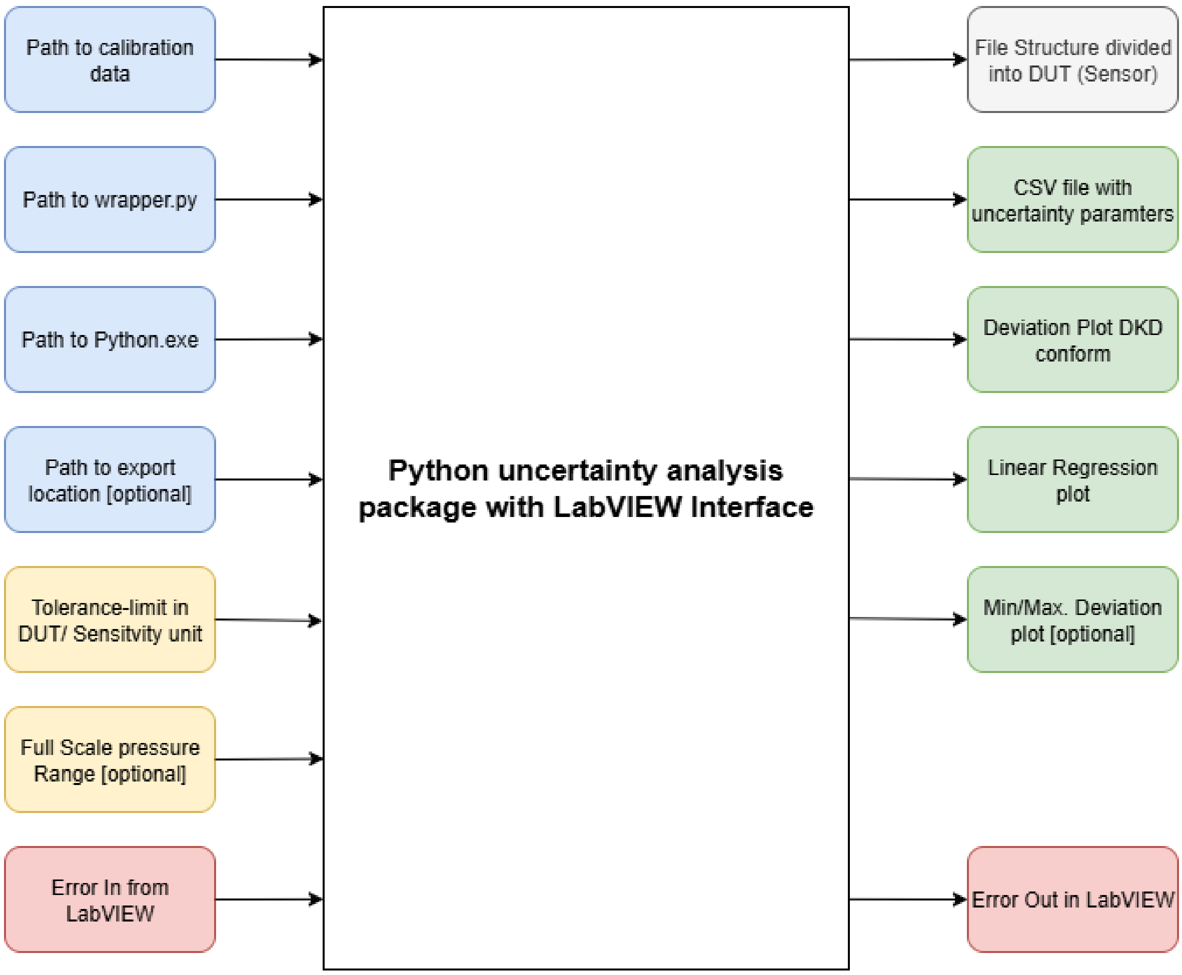

Figure 3 Inputs and Outputs

Inputs

As shown on the left side of the flowchart, several input parameters are given, where blue stands for paths, yellow for floats, and red for error propagation:

- **Path to calibration data:** This inputs the location of the CSV file that contains unprocessed calibration data, including the reference pressures and the measurements of the DUTS.
- **Path to wrapper.py:** This points to the path of the wrapper.py file and runs it, connecting the LabVIEW interface with the Python uncertainty analysis package.
- **Path to Python.exe:** Required to run the Python.exe from the correct environment.
- **Path to export location (optional):** Optional directory where all generated result files are saved. If there is no input, a default preset path will be chosen.
- **Tolerance limit in DUT/Sensitivity unit:** This float defines the acceptable deviation range. Given either in Pascal for digital sensors or as Sensitivity (Volt/Pascal) for analog sensors.
- **Full-scale pressure range (optional):** If specified, this value is used for the plot to get the deviation as a percentage of the Full-Scale pressure range.
- **Error In from LabVIEW:** Allows error propagation from earlier LabVIEW errors.

Together, these inputs provide all necessary information for calibrating the post-processing tool.

Outputs

The right side of the flowchart shows the outputs of the Python uncertainty package:

- **File structure divided into DUTs:** The results are grouped by DUT and stored in a folder with the DUT number of the sensor.
- **CSV file with uncertainty parameters:** A CSV result file is created for each DUT, containing different uncertainty contributors and other related numerical outputs.
- **Deviation plots (DKD-R 6-1-compliant):** Include sensitivity plots for analog sensors or pressure deviation plots for digital DUTs, each with corresponding uncertainty bounds and tolerance limits.
- **Linear regression plot:** Shows the relationship between reference pressure and DUT output, used internally for evaluating linear response characteristics.

- **Optional min/max scatter plots:** Visualize maximum measurement spread per pressure level, provided full-scale pressure range is specified.
- **Error Out in LabVIEW:** Return errors from earlier LabVIEW errors and any error that might have occurred during the uncertainty analysis.

2.3.2. Software Architecture

Software is divided into three main parts. As shown in the figure 4 below, the program consists of a LabVIEW function that can be connected to the existing LabVIEW calibration routine. The LabVIEW function provides all inputs and calls the wrapper.py module, which connects the uncertainty analysis package and the LabVIEW function, which calls the uncertainty analysis package. The uncertainty analysis package does all the calculations and generates all of the results.

Figure 4 Flowchart of the software architecture with the three main components

Python Architecture

The task was to program a package in Python that calculates the different uncertainty contributors, expanded relative and absolute uncertainty, and other related parameters to automate the calibration. While being DKD-R 6-1 compliant in every module of the code, for the calculations given by the DKD-R 6-1 earlier. Also, at the end, a Python package should be created that can be exported and used on any computer.

A modular programming structure was implemented, where each uncertainty contributor has its own module and follows a functional programming style with primarily encapsulated functions. This design enhances maintainability and facilitates development, as errors in one module do not affect the functionality of the entire tool.

A constants module was defined to allow easy modification of key variables. This module utilized enum classes to store constants, such as the relative uncertainty from the reference device and the number of measurement series. Therefore, simplifying the code structure.

Additionally, several helper functions were implemented. For example, one function identifies the zero index in the measurement series based on the reference pressure, which is required for multiple subsequent calculations. Another function computes linear regression parameters used in sensitivity analysis.

Each uncertainty contributor's hysteresis, zero deviation, repeatability, and mean pressure values were calculated to be fully compliant with DKD-R 6-1, in absolute values for digital and as a percentage for analog sensors. The other uncertainty contributors, resolution and reference device uncertainty, were calculated relatively for each pressure level, and the expanded relative uncertainty was calculated from there in the uncertainty module of the package.

The Python package structure is shown in figure 8:

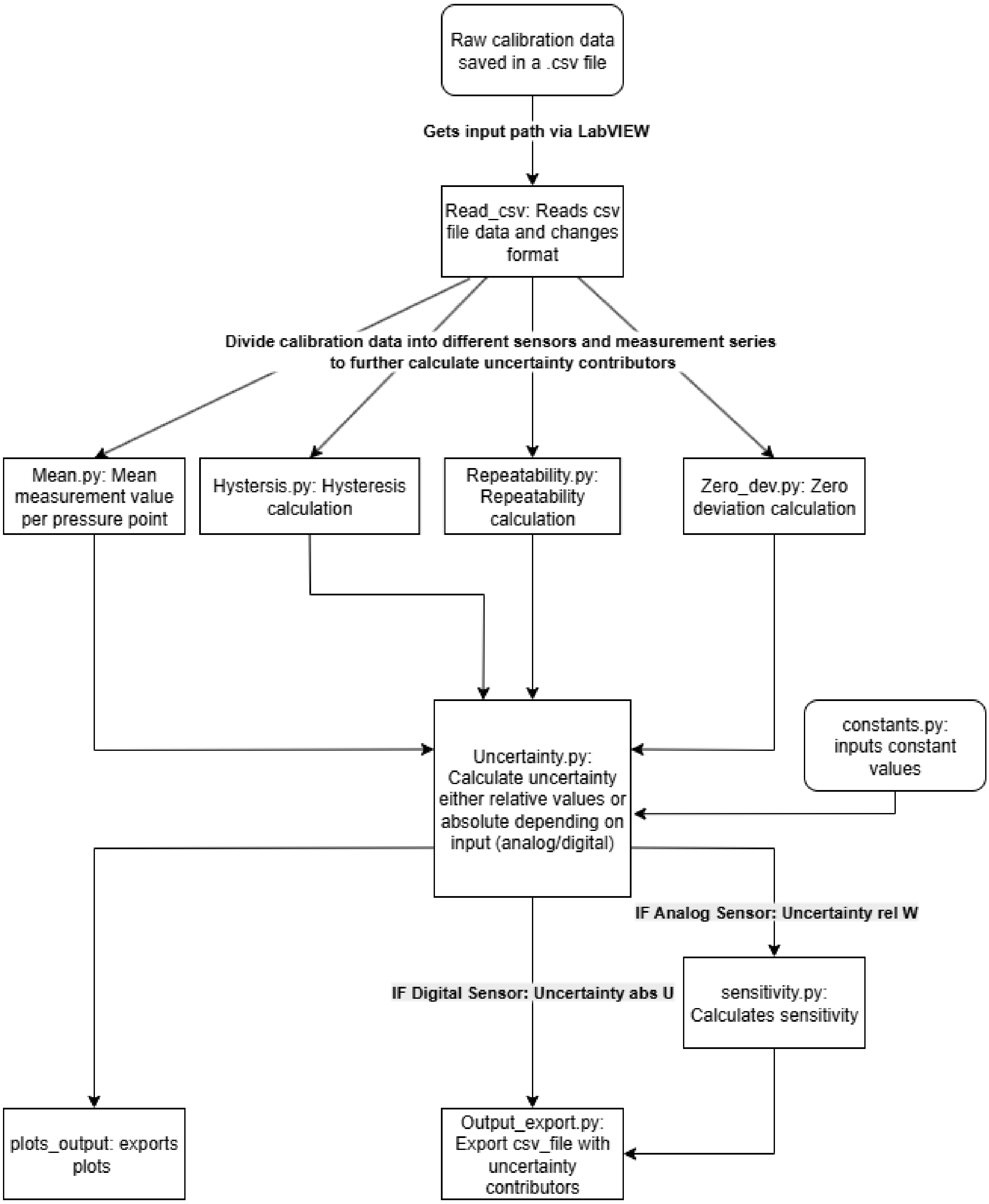

Figure 5 Schematic of the structure of the Uncertainty analysis package workflow.

After being read into the package, the raw calibration data is checked for completeness and data integrity for both sensor types in the read_csv.py module. The data is divided by

DUTs and measurement series, which are automatically detected. Afterwards, these measurement series are called by multiple calculation modules. The module mean.py, which calculates the zero-point corrected mean value across a DUT.The modules hysteresis.py, repeatability.py, and zero_dev.py calculate the uncertainty contributors as mentioned in equations 2.14-2.18. After the computation, these values and constant uncertainties from constants.py are read into uncertainty.py and divided by distribution type (A factor of three for a rectangular distribution, or two for a normal distribution). Following this, either the relative measurement uncertainty contributors are calculated for analog sensors or the absolute measurement uncertainty contributors are calculated for digital sensors. Then the calculation is divided by sensor type. Analog Sensor data goes to the sensitivity.py module, where the sensitivity parameters are calculated as mentioned in section 2.2.4 and then are provided to the output_export.py. The uncertainty parameters are recorded in a CSV file in a company internal format for both sensor types. Additionally, the data also goes to the plots_output.py module. This module generates the plots from the provided data and outputs them as PNG files.

The documentation was built with the Sphinx library [17], which takes information from docstrings written for every function. The docstrings are structured into inputs, outputs, and datatypes, then written into an HTML file.

The build library in Python was used to export the package. That created a wheel that another user can pip install and use via importing "pressure_sensor". Then, in Python, call python wrapper.py 'c:`to`calibration_data.csv

The README file gives instructions on how to set up the Python package. Due to compatibility issues between the existing LabVIEW setup, Python version 3.9-32-bit must be installed. The program can then be run using a local Venv (virtual environment). Afterwards, internal wheels can be installed for error handling, and a special layout for the CSV file will be generated. Then the package uncertainty_analysis_python for both sensor types can be installed.

An issue arose because the existing LabVIEW version was only compatible with Python version 3.9.0, 32-bit. A wheel was created to make the package usable on another computer, the paths and imports from local to global were changed, and Python was callable from LabVIEW. To solve those issues, a Python module called wrapper made the package callable from LabVIEW. Another issue was correctly importing and structuring the module to be called on any computer, and installing the package. Therefore, the imports had to be changed from absolute to relative imports, and the wrapper function was moved out of the package to maintain user-friendliness.

LabVIEW Architecture

A LabVIEW VI (Virtual Instrument) embedded in the existing calibration routine that calls the Python-based uncertainty analysis tool was created.

Two implementation options were considered: a LabVIEW class or a LabVIEW VI (similar function). The latter was chosen because the VI was simpler to implement and could be easily integrated into the existing calibration workflow. The VI was packaged into a LabVIEW library to protect the core logic. This ensures it can be reused independently and prevents accidental modification by end users.

The VI serves as a modular wrapper that forwards all required parameters to a Python method and executes the post-processing routine. Once called, the Python backend performs the uncertainty calculation, generates result plots and tables, and stores them either in a user-defined directory or a default location. The VI is designed for minimal user interaction and smooth automation within the LabVIEW-based calibration interface. Moreover, LabVIEW also outputs a zero for success or an error if the calibration fails.

Some problems encountered include graphical programming in LabVIEW, which differs significantly from traditional development environments. The main challenge was establishing a reliable connection between LabVIEW and Python using the **wrapper.py** module. Special attention had to be paid to path formatting, interpreter compatibility, and system execution handling.

2.3.3. Testing and Verification

A multi-layered testing approach was used to verify the functionality and correctness of the software, including artificial calibration data, unit tests, and integration tests.

Synthetic calibration data

Two synthetic datasets were created since no real calibration datasets were available during development. One dataset contained data for a single DUT to allow for faster testing, while the second included multiple DUTs to verify the handling of multiple inputs.

The datasets were modified to have both valid and invalid data. Invalid entries, such as missing headers or incorrect data types, were deliberately used to check the robustness and error handling of the software.

Unit testing

Every function was tested independently to verify that all parts of the software tool work correctly. The pytest [18] framework was used for these unit tests.

Tests are based on manually created datasets that mimic valid calibration scenarios and intentionally invalid inputs. The tests check each function's robustness, including faulty data such as missing values.

Example Test
The following unit test provides a simplified example that runs the function **calculate_-zero_deviation_multiple_cycles** with simplified test data and uses assertions to check if the output matches the expected results.

```
def test_single_cycle_correct_deviation():
    # ref has zero at index 2
    ref = np.array([5, 1, 0, 2])
    up = np.array([10, 12, 20, 15])
    down = np.array([8, 11, 25, 13])
    matrix = np.column_stack((ref, up, down))

    result = calculate_zero_deviation_multiple_cycles(matrix)

    assert "f0_max" in result
    assert "cycles" in result
    assert len(result["cycles"]) == 1
    assert result["f0_max"] == 5  # 25 - 20
```

Furthermore, an important technique called **mocking** was used to simulate external dependencies, such as functions and inputs. Mocking allowed these tests to run independently, making each function testable. Furthermore, many tests explicitly use assert statements, simple checks verifying that the code output matches expected results.

Additionally, test coverage was measured using the **coverage** library [19] to identify any untested parts of the code. In figure 6 below, an example output of a coverage report is displayed, which shows which lines of code were covered by the tests. In the figure 6 below is an example output of a coverage test report.

```
(benv39_32) PS C:\Users\Raed\Thesis_Raed\Pressure-Sensor-Calibration> coverage report -m
Name                                      Stmts   Miss  Cover   Missing
-----------------------------------------------------------------------
src\pressure_sensor\__init__.py               0      0   100%
src\pressure_sensor\constants.py             33      0   100%
src\pressure_sensor\helpers.py               22      3    86%   58-60
src\pressure_sensor\hysteresis.py            47     14    70%   87-103
src\pressure_sensor\mean.py                  40      7    82%   70-77, 81
src\pressure_sensor\output_export.py         90     14    84%   201, 222, 246-247, 280-284, 296-305
src\pressure_sensor\plots_output.py         175     39    78%   105, 109-110, 140, 151-165, 213-214, 217-218, 226-227, 233-234, 313-314, 386-400, 412-420
src\pressure_sensor\read_csv.py              96     21    78%   69, 71, 78, 158, 173-193
src\pressure_sensor\rel_uncertainty.py      106     15    86%   115-116, 196-197, 221, 225-241
src\pressure_sensor\repeatability.py         36     13    64%   73-92
src\pressure_sensor\sensitivity.py           46     17    63%   101-136, 140
src\pressure_sensor\zero_dev.py              33     13    61%   52-72
tests\__init__.py                             0      0   100%
tests\test_constants.py                       0      0   100%
tests\test_helpers.py                        43      0   100%
tests\test_hysteresis.py                     42      0   100%
tests\test_mean.py                           29      0   100%
tests\test_output_export.py                  83      0   100%
tests\test_plots_output.py                   62      0   100%
tests\test_read_csv.py                       99      0   100%
tests\test_rel_uncertainty.py                68      0   100%
tests\test_repeatability.py                  17      0   100%
tests\test_sensitivity.py                    42      0   100%
tests\test_wrapper.py                        57      0   100%
tests\test_zero_dev.py                       26      0   100%
wrapper.py                                   48     10    79%   70-71, 115, 122-129
-----------------------------------------------------------------------
TOTAL                                      1340    166    88%
```

Figure 6 Coverage test report of uncertainty analysis package

This testing approach ensured the software handled errors gracefully and made the code easier to develop and maintain. Through these tests, any errors introduced during changes in the code could be identified and located more easily, saving time in debugging.

Integration Tests

Integration tests were performed to evaluate the Python-LabVIEW Software tool as a whole. This means that the whole setup was called via LabVIEW. The synthetic calibration data was processed, and the tool generated the uncertainty contributors and various plots. This was done on the development device and a test rig computer to ensure consistency in the results with different virtual environments.

2.4. Results

2.4.1. Automated Output and Workflow Integration

The developed software enables a fully automated calibration output process. Once initiated via a LabVIEW virtual instrument (VI) through a system call, the Python routine autonomously performs all required calculations and data processing steps. For each Device Under Test (DUT), a CSV file is generated that contains all relevant uncertainty contributors, as well as regression results such as sensitivity and offset. In addition to numerical output, the routine produces DKD-R 6-1-compliant result plots (e.g., linear curve fit and deviation diagrams) in a PNG image format, which are saved alongside the data. Each DUT has a separate folder containing one DUT.

The system is designed to support multi-sensor calibration, allowing multiple DUTs without additional user input. After the initial trigger, the routine runs entirely without human interaction. The generated data is structured to meet the documentation requirements for DAkkS calibration certificates.

2.4.2. Export files

The output CSV file contains all calculated uncertainty contributors and regression results per pressure level.

Table 6 provides an overview of the column structure and their descriptions:

Table 6 CSV export column fields with their respective description.

Label [Unit]	Description
Idx [–]	Measurement index
REF [Pa]	Nominal pressure from reference device
DUT [Pa]	Output signal measured by DUT (Device Under Test)
ΔP [Pa]	Difference between DUT and reference
U_r [Pa]	Uncertainty from resolution (absolute)
U_{ref} [Pa]	Uncertainty from reference device
U_{f0} [Pa]	Uncertainty from zero offset deviation
U_b [Pa]	Uncertainty from repeatability
U_h [Pa]	Uncertainty from hysteresis
U_{tot} [Pa]	Expanded total uncertainty
w_i^2 [Pa^2]	Variance of uncertainty
Sens. [Pa/Pa]	Regression slope
Offset [Pa]	Regression offset
StdDev [Pa]	Standard deviation of residuals
R [–]	Correlation coefficient (dimensionless)

An example output from a DUT calibrated against the reference device CPC6050 is shown in Table 7.

Table 7 Example Measurement Data: DUT 1, Reference Device CPC6050

Idx	REF	DUT	ΔP	U_r	U_{ref}	U_{f0}	U_b	U_h	U_{tot}	w_i^2	Sens.	Offset	StdDev	R
0	-1000.53	-1008.19	-7.66	1e-6	0.0001	0.2690	8.5566	4.4216	19.2706	92.8390	1.0032	-2.3074	1.7266	0.999998
1	-499.43	-501.50	-2.07	1e-6	0.0001	0.2690	2.9910	2.4543	7.7568	15.0419	1.0032	-2.3074	1.7266	0.999998
2	0.50	-0.211	-0.711	–	–	–	–	–	–	–	1.0032	-2.3074	1.7266	0.999998
3	500.43	499.61	-0.816	1e-6	0.0001	0.2690	0.1100	0.2911	0.8228	0.1692	1.0032	-2.3074	1.7266	0.999998
4	1000.43	1000.16	-0.272	1e-6	0.0001	0.2690	0.1036	0.0283	0.5794	0.0839	1.0032	-2.3074	1.7266	0.999998

Measurements of reference values at five different pressure levels and the measured DUT value are shown. The measurement uncertainty is absolute because the input is from a digital sensor (from the DUT unit Pascal). For analog sensors, the output differs; instead of absolute measurement uncertainties, the values are relative to the DUT output, and sensitivities mentioned in the calculations section are displayed as well. Furthermore, the expanded measurement uncertainty of the sensitivity and the sensitivity's error span are also shown.

The linear fit plot figure 7 displays the linear curve fit of the calibration data as DUT value over reference pressure, as well as statistical parameters for the data. The higher the correlation, the better the sensor, allowing for the assessment of sensor linearity visually and numerically.

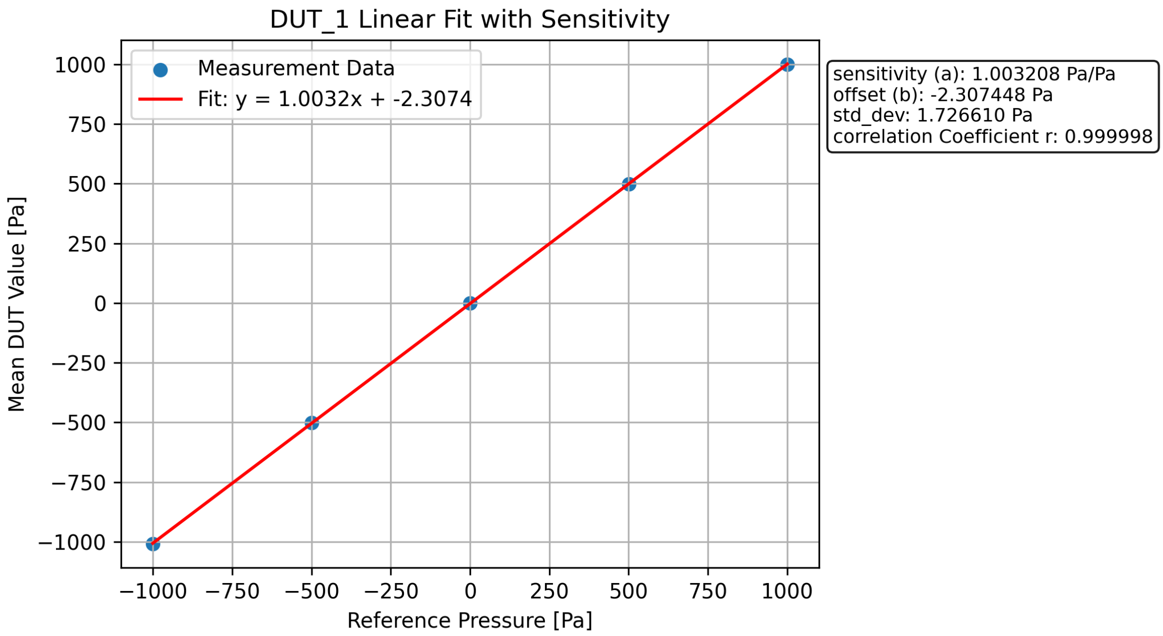

Figure 7 Linear curve fit plot of a digital sensor. The blue dots show the measurement data, while the red line is the resulting linear curve fit. The fitting parameters and the correlation coefficient can be seen in the table on the top right of the plot.

The figure 8 shows another plot that is generated. This plot is needed for the calibration certificate and either displays expanded measurement uncertainty and deviation in Pascal for digital sensors, as shown in the graphic below, or displays measurement uncertainty of the sensitivity for analog sensors.

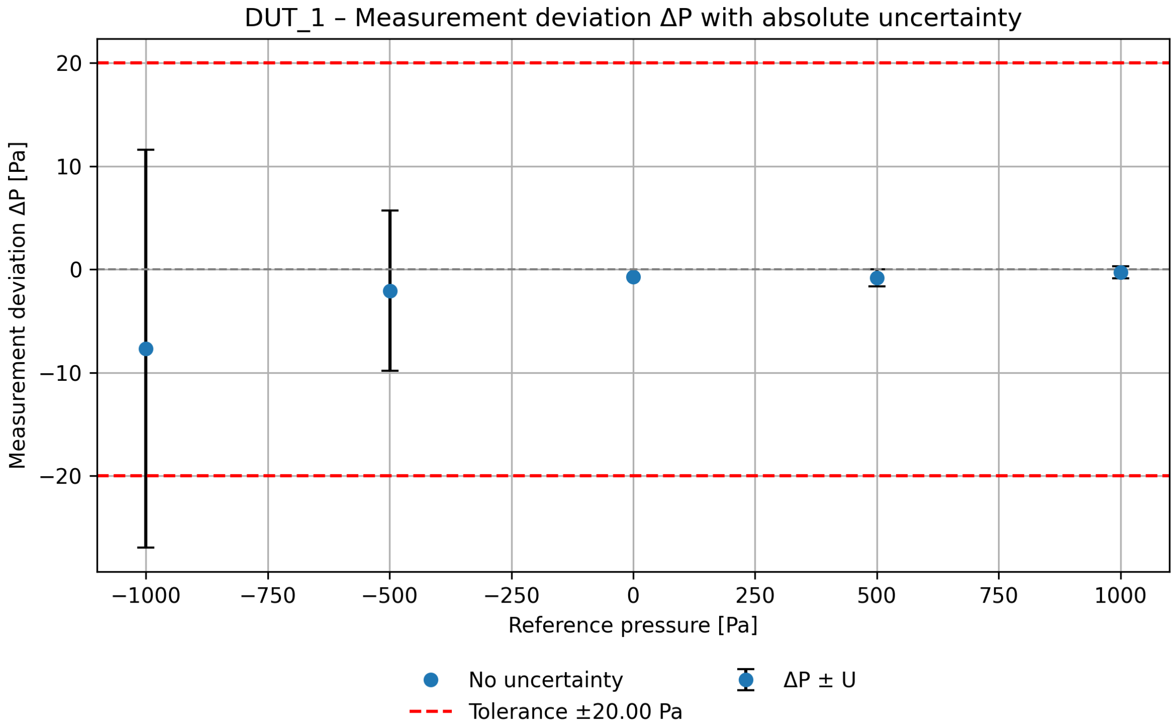

Figure 8 Example of a DKD-R 6-1 conform uncertainty deviation plot. That is generated by the implemented tool. The blue dots show the average measurement deviation per pressure level, while the black error bars show the expanded measurement uncertainty span. The red dashed line shows the tolerance limit in Pascal.

The figure 9 shows the deviation and tolerance limit as a percentage of the sensor's Full Scale and plots every measurement point. Allowing for an assessment of whether any measurement point or series had a significant deviation. Therefore, allowing an easy visual quality assessment.

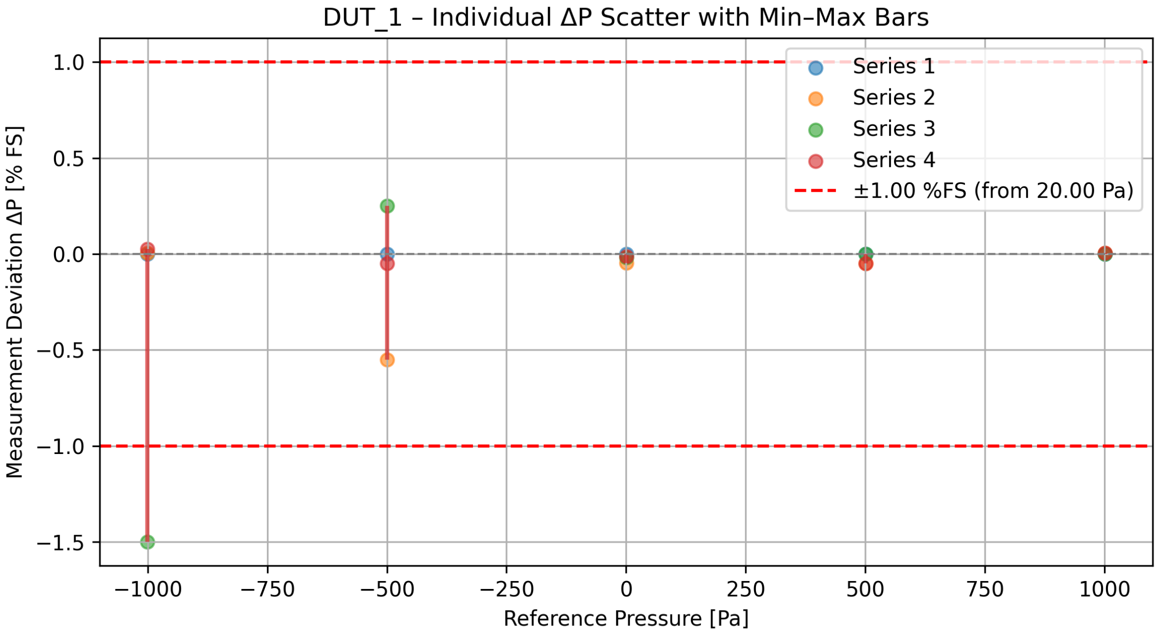

Figure 9 The plot shows the tolerance limit and deviation as a percentage of the sensor's full-scale range. The plotting of each measurement point and series allows for assessing the quality of the calibration.

3. Conclusion and Outlook

3.1. Conclusion and Critical Reflection

3.1.1. Conclusion

This thesis aimed to develop an automated software tool for the uncertainty evaluation of pressure sensors in compliance with the DKD-R 6-1 guideline and ISO/IEC 17025, and to compute calibration coefficients. The tool was implemented using Python for the backend processing and LabVIEW for frontend interaction, enabling seamless integration into the existing calibration routine at Vectoflow.

The resulting software tool successfully automates key steps in the uncertainty analysis process, including repeatability, hysteresis, and zero deviation calculation. Eliminating manual calculation steps significantly reduced time spent on calibration, increased the consistency of the results, and improved traceability by eliminating human errors from the uncertainty calculations.

3.1.2. Critical Reflection

However, some limitations need to be discussed to critically evaluate the scope and performance of the developed software tool.

A central limitation lies in the uncertainty analysis framework itself. The tool was developed strictly in compliance with the DKD-R 6-1 guideline, which mandates the use of deterministic methods such as the root-sum-square (RSS) model for combining uncertainties. As a result, more advanced approaches such as Monte Carlo simulations (MCM), which are gaining relevance in modern metrology, could not be implemented. Monte Carlo methods allow for more realistic modeling of uncertainty, especially in complex or non-linear systems.

Another technical limitation was the existing LabVIEW infrastructure at Vectoflow. To ensure compatibility with the LabVIEW Python Node, the software had to be developed and tested using Python version 3.9, 32-bit. This restricted the choice of external libraries and required careful management of dependencies, particularly when packaging the software for portability.

Additionally, although the tool was thoroughly tested using synthetic calibration data and a comprehensive unit test suite, further testing is necessary to cover more edge cases.

3.2. Outlook

Figure 10 presents an overview of the complete calibration process.

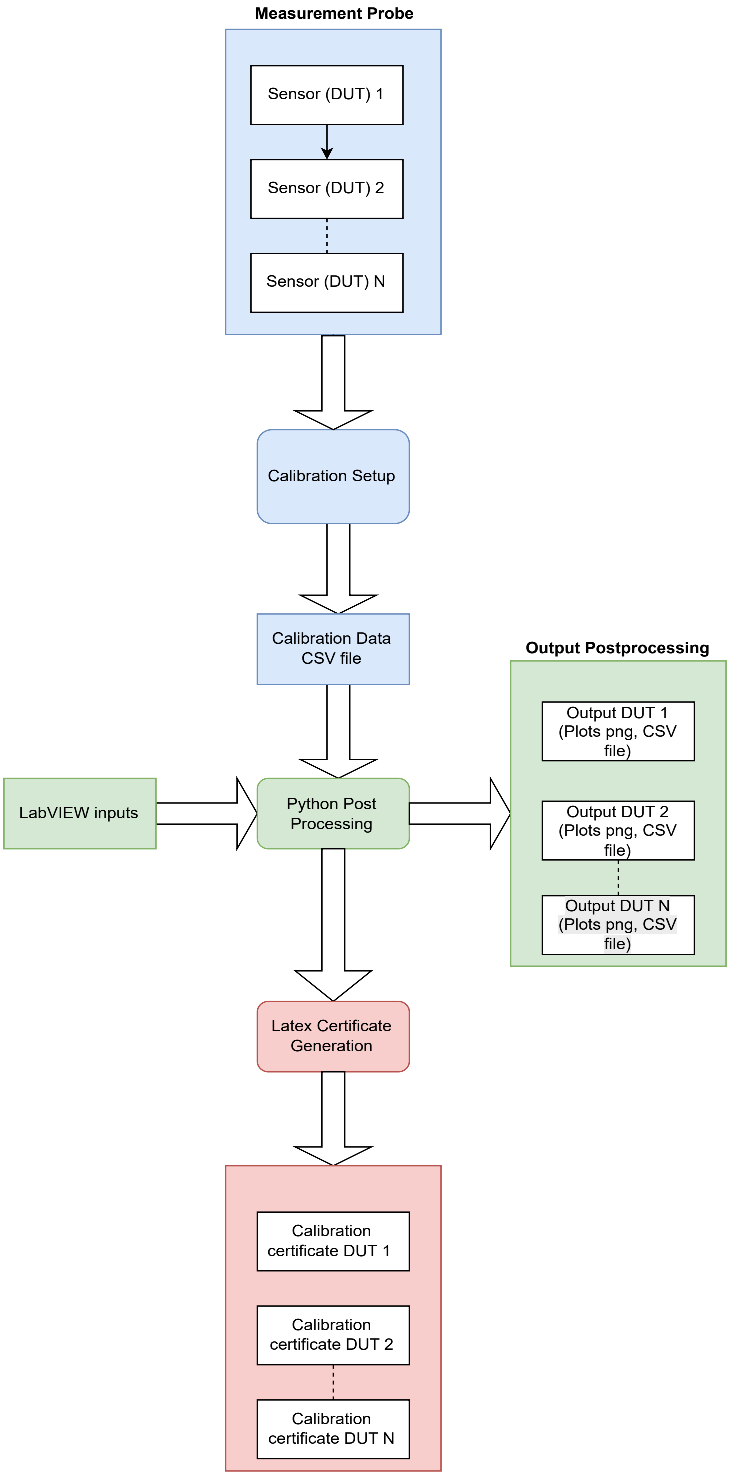

Figure 10 The figure shows the end-to-end calibration sequence: existing components are marked in blue, implementations completed during this thesis are shown in green, and potential future extensions are highlighted in red.

The components marked in green were implemented as part of this thesis. Preexisting modules are highlighted in blue. Components marked in red represent potential future extensions.

An automatic generation of calibration certificates based on the Python post-processing outputs will be created in the future. Since the current implementation already computes all values required for certificate creation, this step could be realized by integrating a LaTeX-based reporting system. The LaTeX script could be invoked directly from Python, enabling fully automated, DKD-R 6-1-conform certificate generation. Once verified, this would allow the complete calibration workflow to be DAkkS-accredited.

The tool's modular structure and consistent performance provide a solid foundation for extensions, such as certificate or full workflow automation. These additions have the potential to transform the system into a fully digitalized calibration solution for industrial applications.

Bibliography

[1] Mensor, “The fundamentals of pressure calibration,” Online PDF document, 2021, accessed: 2025-07-24. [Online]. Available: https://cdn2.hubspot.net/hubfs/3994797/The%20Fundamentals%20of%20Pressure%20Calibration.pdf?__hstc=22053220.d209b579e7b870a0ad99a0fc9e228b44.1752329820624.1752329820624.1752394632204.2&__hssc=22053220.4.1752394632204&__hsfp=2053954935

[2] I. O. for Standardization, “Iso/iec 17025:2017 - general requirements for the competence of testing and calibration laboratories,” International Organization for Standardization, 2017, accessed: 2025-07-31. [Online]. Available: https://www.iso.org/standard/66912.html

[3] German Calibration Service (DKD), “Calibration of pressure gauges,” DKD-R 6-1, Rev. 2, Ed. 03/2014, PTB, Braunschweig, Germany, 2014, accessed: 2025-05-27. [Online]. Available: https://www.ptb.de/cms/fileadmin/internet/dienstleistungen/dkd/archiv/Publications/Guidelines/DKD-R_6-1_2016_englisch.pdf

[4] F. Corporation, “Pressure comparators and digital reference pressure gauges vs deadweight testers,” 2023, accessed: 2025-05-27. [Online]. Available: https://www.fluke.com/en-us/learn/blog/pressure-calibration/pressure-comparators-and-digital-reference-pressure-gauges-vs-deadweight-testers

[5] M. Zou, Y. Xu, J. Jin, M. Chu, and W. Huang, “Accurate nonlinearity and temperature compensation method for piezoresistive pressure sensors,” *Sensors*, vol. 23, no. 13, p. 6167, 2023, accessed: 2025-07-24.

[6] “Pace6000 precision pressure controller,” https://calibration-services.calcert.com/Asset/PACE6000_Datasheet.pdf, accessed: 2025-05-27.

[7] Advanced Energy, “PS-CAL Software Calculations Explained.” [Online]. Available: https://www.advancedenergy.com/en-us/about/news/blog/ps-cal-software-calculations-explained/

[8] N. N. I. of Standards and Technology), “Nist special publication 250-39: Calibration services for pressure using piston gauge standards,” 2009, accessed: 2025-05-25. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication250-39.pdf

[9] JCGM 100:2008, *Evaluation of Measurement Data — Guide to the Expression of Uncertainty in Measurement*. Joint Committee for Guides in Metrology (JCGM),

2008, accessed: 2025-07-15. [Online]. Available: https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf

[10] I. Smith, Y. Luo, and D. Hutzschenreuter, "The storage within digital calibration certificates of uncertainty information obtained using a monte carlo method," *Metrology*, vol. 2, no. 1, pp. 33–45, 2022, accessed: 2025-07-24. [Online]. Available: https://doi.org/10.3390/metrology2010003

[11] F. Hughes, M. Marschall, G. Wübbeler, G. Kok, M. van Dijk, and C. Elster, "Jcgm 101-compliant uncertainty evaluation using virtual experiments," *arXiv preprint arXiv:2404.10530*, 2024, accessed: 2025-07-24. [Online]. Available: https://arxiv.org/abs/2404.10530

[12] A. Abacus, "Pressure sensors – design considerations and technology options," White Paper, 2017, accessed: 2025-08-02. [Online]. Available: https://my.avnet.com/wcm/connect/5902b94e-79e9-45b4-a044-ae932a7facb7/AvA-Pressure-Sensors-EN-White-Paper.pdf?MOD=AJPERES&CVID=oM.stoZ

[13] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2018.

[14] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020, version 1.26, Accessed: 2025-07-24.

[15] W. McKinney, "Data structures for statistical computing in python," *Proceedings of the 9th Python in Science Conference*, pp. 51–56, 2010, version 1.5.3, Accessed: 2025-07-24.

[16] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, version 3.7.1, Accessed: 2025-07-24.

[17] G. Brandl and S. contributors, "Sphinx: Python documentation generator," https://www.sphinx-doc.org/, 2024, version 7.4.7, Accessed: 2025-07-24.

[18] H. Krekel and pytest contributors, "pytest: Simple powerful testing with python," https://docs.pytest.org/, 2024, version 8.4.1, Accessed: 2025-07-24.

[19] N. Batchelder, "Coverage.py," https://coverage.readthedocs.io/, 2024, version 7.9.1, Accessed: 2025-07-24.

[20] AMETEK Calibration, "Deadweight vs gauge pressure calibration white paper," AMETEK Calibration, Tech. Rep., accessed: 2025-05-28. [On-

line]. Available: https://www.ametekcalibration.com/-/media/ametekcalibration/download_links/white-papers/deadweight-vs-gauge-white-paper-us.pdf?la=en&revision=05aaa91c-5f94-497b-bf19-44919255901f&hash=64070F2D3D729049D800ED4105E40F80

[21] T. Henderson. (2025) Understanding iso/iec 17025 requirements for measurement uncertainty (mu). Accessed: 2025-06-02. [Online]. Available: https://www.labmanager.com/uncertainty-in-measurement-training-program-16756

Tools and Aids Used

To support the preparation and quality of this thesis, the following tools were employed:

- **Grammarly** – for automated grammar and style suggestions to improve English language clarity.
- **DeepL** – for translating initial drafts and technical terms from German to English.
- **Latex** – for writing and formatting the thesis

Declaration

I hereby declare that I have written the submitted thesis independently and have used no sources or aids other than those stated.

Gilching, 02.08.2025, Raed Tarar

List of Figures

Figure 1. Calibration Setup with respective measurement uncertainty contributors. Marked in red are environmental conditions, which do not contribute to the measurement uncertainty but need to be stable for a successful calibration. 8
Figure 2. Visualization of the calibration sequence A (p. 13 [3]) 9
Figure 3. Inputs and Outputs 22
Figure 4. Flowchart of the software architecture with the three main components 24
Figure 5. Schematic of the structure of the Uncertainty analysis package workflow. . 26
Figure 6. Coverage test report of uncertainty analysis package 30
Figure 7. Linear curve fit plot of a digital sensor. The blue dots show the measurement data, while the red line is the resulting linear curve fit. The fitting parameters and the correlation coefficient can be seen in the table on the top right of the plot. 33
Figure 8. Example of a DKD-R 6-1 conform uncertainty deviation plot. That is generated by the implemented tool. The blue dots show the average measurement deviation per pressure level, while the black error bars show the expanded measurement uncertainty span. The red dashed line shows the tolerance limit in Pascal. 34
Figure 9. The plot shows the tolerance limit and deviation as a percentage of the sensor's full-scale range. The plotting of each measurement point and series allows for assessing the quality of the calibration. 35
Figure 10. The figure shows the end-to-end calibration sequence: existing components are marked in blue, implementations completed during this thesis are shown in green, and potential future extensions are highlighted in red. 37

List of Tables

Table 3. Calibration Sequences according to DKD-R 6-1 p. 12 table 1 [3] 9
Table 4. Analog and Digital Sensor comparison 11
Table 5. Uncertainty Contributors in Pressure Sensor Calibration 16
Table 6. CSV export column fields with their respective description.......................... 31
Table 7. Example Measurement Data: DUT 1, Reference Device CPC6050 32