TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik

CSR-25-09

Development of an Automation Framework for 1D Measurement

Alhassan Khalil · Reda Harradi · Stephan Rupf · Wolfram Hardt

Dezemberr 2025

Chemnitzer Informatik-Berichte

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Development of an Automation Framework for 1D Measurement

Master Thesis

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc. Automotive Software Engineering

Dept. of Computer Science
Chair of Computer Engineering

Submitted by: Alhassan Khalil
Student ID: 730044
Date: 18.08.2025
Supervising Tutor: Prof. Dr. Dr. h. c. Wolfram Hardt
External Supervisor: Stefan Rupf

Acknowledgements

I would like to express my sincere gratitude to Mr. Stefan Rupf, my supervisor at 3D-Micromac AG, for his exceptional guidance, technical expertise, and unwavering support throughout the course of this thesis. His constructive feedback, attention to detail, and commitment to high standards have been invaluable in shaping both the quality of the work and my professional growth

My appreciation also goes to Mr. Torsten Siadle, my team leader, for his constant encouragement, practical insights, and readiness to share his experience, which greatly contributed to the successful completion of this project.

Finally, I wish to express my heartfelt thanks to my family for their unconditional love, endless encouragement, and belief in my abilities, which have been a constant source of strength and motivation. I am also deeply grateful to Lama Mansour for her patience, understanding, and continuous encouragement during this journey.

Abstract

Modern laser micromachining machines must move with very high accuracy to keep product quality consistent. At 3D-Micromac, the earlier calibration process used several separate tools: one for moving the axis, another for reading the laser interferometer, and a spreadsheet for the calculations. This separation made the work slow, error-prone, and hard to repeat.

This thesis builds a single, automated framework that joins all steps from axis motion, data capture, error calculation, and report creation inside the company's microSTAGE software. The framework has four main parts Motion Control, Measurement, Analysis, and Reporting. The different parts can communicate with each other seamlessly thanks to standard file formats and clear programming interfaces. Python handles the maths and plots. Ready-made HTML templates turn the results into calibration certificates that follow the relevant international standard.

Tests on a 200 mm linear axis show that the new tool matches the results of the old spreadsheet and more accurate while cutting total calibration time from about two hours to under twenty minutes. After applying the correction table produced by the framework, the overall positioning error fell by roughly one-third.

The new system removes manual file handling, keeps a clear record of every data point, and can easily grow to cover more axes, different sensors, or future standards making 3D-Micromac's machines easier to calibrate and more reliable. **Keywords: axis calibration, laser interferometry, ISO 230-2, automation framework, measurement software**

Table of Content

Acknowledgements . . . . **i**

Abstract . . . . **ii**

Table of Content . . . . **iii**

List of Figures . . . . **vi**

List of Tables . . . . **viii**

List of Abbreviations . . . . **ix**

1. Introduction **1**

1.1. Motivation 1
1.2. Problem Statement 2
1.3. Objectives 2
1.4. Scope and Limitations 3
1.5. Structure of the Thesis 3

2. Theoretical Background **5**

2.1. Motion Control Systems in Laser Micromachining 5
2.2. Structure of Positioning Systems 6
2.2.1. Positioning Sensors 7
2.3. Principles of Dimensional Metrology 8
2.4. Motion Control in Measurement Systems 11
2.5. Interferometry Fundamentals 14
2.6. Standards for Evaluating Axis Performance 17
2.7. Automation and Modularity in Measurement Systems 19
2.8. Python in Metrology Automation 20
2.8.1. Rationale for Adoption in Measurement Science 20
2.8.2. Scientific Computing Stack for Metrology 21
2.8.3. Interfaces to Industrial Measurement Hardware 21
2.8.4. Implications for One-Dimensional Calibration Frameworks . . 22

3. State of the Art **23**

3.1. Current Workflow at 3D-Micromac 23
3.2. Calibration Techniques for Automated 1D-Measurement 25

3.3. Motion Subsystems for High-Precision 1D Measurement 27
3.4. Laser Interferometry Sensors 29

4. Concept 31
4.1. Design Overview 31
4.2. System Components 33
4.2.1. Motion Control Module 33
4.2.2. Measurement Module 34
4.2.3. Analysis Module 35
4.2.4. Man-Machine Interface (MMI) 37
4.2.5. Reporting Module 39
4.3. Data Flow and Communication 41
4.4. Modularity and Scalability 43

5. Implementation 45
5.1. System Overview 45
5.2. Motion Control Integration 47
5.3. Data Acquisition Integration 50
5.4. Data Analysis 51
5.4.1. Metric Calculations 52
5.4.2. Visualization 54
5.4.3. Output Structure 55
5.5. User Interface (MMI Integration) 57
5.5.1. User Interaction and Workflow 57
5.5.2. Result Display and Visualization 59
5.6. Measurement Protocol 61

6. Results and Evaluation 63
6.1. Experimental Setup 63
6.1.1. Hardware Configuration 63
6.1.2. Environmental Control and Monitoring 64
6.1.3. Motion Programme 65
6.1.4. Data-acquisition Chain 66
6.2. Performance Metrics 66
6.3. Data and Observations 68
6.4. Evaluation of Accuracy and Repeatability 73

7. Discussion 75
7.1. Summary of Key Outcomes 75
7.2. Limitations and Technical Constraints 76

8. Conclusion and Future Work 79
8.1. Conclusion 79
8.2. Future Work 80

Bibliography **81**

Bibliography from TUC **90**

A. WorkFlow diagrams **91**

B. Sample Protocol Reports before calibration **94**

C. Sample Protocol Reports after calibration **100**

List of Figures

2.1. Illustration of the structure of laser processing systems [2]. 6
2.2. Classification of Position Sensors 7
2.3. Accuracy vs. Repeatability (Precision)[16]. 9
2.4. diagram shows how backlash can happen in gear systems[20]. 10
2.5. Linear and ball-screw drive mechanisms [23]. 11
2.6. Block diagram of process control using PID [24]. 12
2.7. Cascade control for feed controls [30] 13
2.8. Michelson interferometer setup showing beam paths and resulting interference fringes. Mirror displacement causes fringe shifts proportional to the optical path change[36]. 14
2.9. A simple view of the quadrature detection [41]. 15
2.10. Triple beam interferometer measurement principle [43]. 16
2.11. Standard test cycle 17

3.1. Metric gauge block set 25

4.1. Current Workflow 32
4.2. Expected Workflow 32
4.3. User input and measurement execution flow in the MMI interface. . . 38
4.4. Post-measurement logic for calibration and protocol handling. 39
4.5. Data Flow Architecture of the 1D Measurement Automation System . 42

5.1. Hardware architecture and communication layout of the measurement system. The PC (IPC) runs microSTAGE and controls the motion system via EtherCAT and reads interferometer data via USB. 46
5.2. Overview of the MMI implementation showing configuration/display panel (1), action buttons (2), log console (3), and axis status display (4) 57
5.3. an example of the configuration panel in micrSTAGE. 58
5.4. Result display interface in microSTAGE showing metric table and visual graphs 60

6.1. Experimental rig comprising an AeroTech PRO165 vertical axis, a SIOS SP 2000 TR interferometer head, and the host PC running microSTAGE. 64
6.2. Pendulum motion pattern, the axis visits target positions in a forward–backward pair,repeated for number of cycles 65

6.3. Bidirectional positioning deviation and repeatability before calibration. 68
6.4. Forward and backward deviation across all five cycles before calibration 69
6.5. Bidirectional positioning deviation and repeatability after calibration 70
6.6. Forward and backward deviation across all five cycles after calibration 70

List of Tables

4.1. ISO 230-2 Evaluation Metrics Used in the Analysis 36

6.1. Comparison of ISO 230-2 metrics before and after calibration 73

List of Abbreviations

ACS	ACS Motion Control (servo controller brand)
API	Application Programming Interface
CNC	Computer Numerical Control
COM	Component Object Model
CSV	Comma-Separated Values
D3	Data-Driven Documents (JavaScript library)
DGQ	Deutsche Gesellschaft für Qualität
EtherCAT	Ethernet for Control Automation Technology
GUI	Graphical User Interface
HTML	HyperText Markup Language
IPC	Industrial Personal Computer
ISO	International Standards Organization
JSON	JavaScript Object Notation
MMI	Man–Machine Interface
PC	Personal Computer
PDF	Portable Document Format
PID	Proportional–Integral–Derivative
PNG	Portable Network Graphics
SIOS	SIOS Meßtechnik GmbH (interferometer vendor)
USB	Universal Serial Bus
VB	Visual Basic
VBA	Visual Basic for Applications
VBScript	Visual Basic Script
VDI	Verein Deutscher Ingenieure
VDI/DGQ	Verein Deutscher Ingenieure/Deutsche Gesellschaft für Qualität

1. Introduction

1.1. Motivation

3D-Micromac engineers and manufactures laser micromachining systems that rely on sub-micrometer positioning precision and sustained repeatability. Even minor axis errors propagate directly into kerf width, taper, or overlay deviations during wafer scribing and other ultra-short-pulse processes, threatening product yield and customer confidence. To guarantee that every machine meets its precision specification, each linear axis must therefore be calibrated, verified, and documented under traceable conditions before a system start production.

At 3D-Micromac, the calibration process has up until now been divided among a number of independent instruments such as the axis controller interface is used to carry out axis motion, the laser interferometer software logs position data, and Microsoft Excel is used to post-process information. This chain of tools results in a lot of manual hand-offs, idle time during file exports and imports, and a high chance of formula or copy-paste errors in spreadsheets. Most critically, repeating the measurement sequence after parameter adjustments requires engineers to restart the entire chain from scratch, slowing iterative optimization and hampering root-cause analysis.

By contrast, a unified automation framework can orchestrate motion control, data acquisition, standards analysis, and protocol generation from within a single microSTAGE application window. Such integration eliminates redundant file transfers, enables scripted data checks, and shortens the feedback loop between measurement and compensation. Preliminary prototypes have already demonstrated how consolidating these steps “replaces what used to be a series of disconnected manual steps with a single, unified automated process,” substantially reducing set-up errors and downtime. In addition, embedding standards calculations directly in the workflow ensures that every certificate produced is standards compliant, comparable, and ready for quality audit trails.

The motivation for this thesis is therefore two fold. first, to raise axis calibration accuracy, and repeatability by eliminating manual gaps. The second is to provide a modular, reusable software platform that internal engineers can extend to new machine types or future metrology standards without rewriting the entire code base. Achieving these goals will strengthen 3D-Micromac’s position as a supplier of high-precision laser systems while reducing commissioning effort on the shop floor.

1.2. Problem Statement

Interferometric calibration is a well-established approach for evaluating the accuracy of linear axis positioning. However, its practical implementation within 3D-Micromac's production workflow has historically been impeded by a fragmented software toolchain. Currently, engineers must manage separate systems: one to control axis movements, another to capture measurement data, and a third to evaluate positioning deviations according to standards such as ISO 230-2. Each of these tools was developed independently, frequently relying on proprietary or vendor specific interfaces that limit seamless data exchange. Consequently, the overall calibration process becomes time-consuming, error-prone, and challenging to standardize.

The lack of integration between these tools introduces several operational inefficiencies. Manual file transfers cause delays and increase the risk of version mismatches or user errors. Spreadsheet-based evaluations depend on complex formulas that are difficult to verify, hard to modify, and easy to corrupt. When adjustments to axis tuning parameters are required, engineers must restart the entire measurement sequence from the beginning, which slows down iterative optimization and prevents real time feedback. Additionally, calibration reports produced through this manual workflow often lack consistency and traceability, raising concerns about their reliability during internal reviews or external audits.

Without a unified software platform that can coordinate axis control, data acquisition, ISO-compliant analysis, and automated reporting, the calibration process remains inefficient and challenging to scale. This fragmentation not only prolongs machine commissioning but also increases variability in measurement results and documentation quality.

This thesis addresses the absence of an integrated, automated calibration framework that ensures process consistency, minimizes human error, and fully aligns with recognized metrology standards. Solving this problem is critical for improving efficiency, guaranteeing reproducibility, and reinforcing 3D-Micromac's commitment to high-precision engineering.

1.3. Objectives

The primary objective of this thesis is to conceive, implement, and validate an integrated automation framework that condenses the entire one-dimensional axis-calibration workflow comprising motion control, interferometric data capture, standards computation, and protocol generation into a single, script-driven module within 3D-Micromac's microSTAGE platform. The framework shall reduce the complete calibration cycle, including setup tasks, from the current duration of approximately two hours to less than thirty minutes by eliminating manual file transfers, spreadsheet manipulation, and repeated operator hand-offs. Although the project does not target a specific numerical tolerance for axis accuracy, the software must perform all standard-compliant calculations automatically so that engineers can quantify deviations, repeatability, backlash, and reversal error directly from the captured data without external tools. By automating every procedural step from axis motion through certificate export the solution is intended to minimise human error, ensure traceable data provenance, and provide a reusable foundation that can be extended to future machine configurations or evolving metrology standards with minimal additional effort.

1.4. Scope and Limitations

This thesis focuses exclusively on the automation of one-dimensional axis calibration using interferometric measurement systems. The developed framework supports the execution, evaluation, and documentation of linear axis motion tests in accordance with standards, including parameters such as positioning deviation (E), repeatability (R), backlash (B), and reversal error (A). The scope includes software side integration with motion controllers and interferometric sensors already in use at 3D-Micromac.

The framework is designed to be modular and scriptable, enabling internal engineers to adapt measurement routines, processing logic, or reporting templates to suit different machine configurations or new calibration standards. Its architecture prioritizes reusability, traceability, and deterministic execution rather than graphical user interaction or broad user configurability.

However, the thesis does not cover multi-axis (2D or 3D) error mapping, real-time hardware in the loop compensation, or integration with external production management systems. The impact of environmental factors such as temperature, humidity and air pressure is taken into consideration solely insofar as they are measured by the interferometer or associated sensors. There is no external environmental correction model in operation. The scope is further limited to the verification phase of axis behaviour motion tuning, and mechanical alignment must be performed separately prior to calibration.

Finally, while the software is developed for internal deployment at 3D-Micromac, its structure and evaluation are tailored to the company's specific hardware stack and process flows. Broader generalization or deployment in other industrial contexts may require adaptation of hardware interfaces or compliance logic.

1.5. Structure of the Thesis

This thesis is structured into eight chapters. Chapter 2 establishes the theoretical foundations of dimensional metrology, interferometry, and standards metrics. Chapter 3 reviews existing automated calibration solutions and highlights the functional gaps that motivate the present work. Chapter 4 outlines the concept and modular architecture of the proposed framework, while Chapter 5 details its implementation within the microSTAGE environment, covering controller communication, data acquisition, analysis routines, and report generation. Chapter 6 presents experimental results obtained on production machines, demonstrating gains in cycle time, repeatability, and documentation quality. Chapter 7 discusses these findings in light of the framework's limitations and potential extensions, and Chapter 8 concludes by summarising the contributions and outlining future research directions. The appendices provide supporting material, such as key source code excerpts and representative calibration logs.

2. Theoretical Background

High-precision axis calibration rests on a chain of physical principles that links actuator motion in the machine tool to traceable length measurements obtained by laser interferometry and, ultimately, to internationally recognised quality metrics. The pages that follow consolidate the foundational material needed later in the thesis. Modern motion-control architectures used in laser micromachining are surveyed, core concepts of dimensional metrology and interferometry are reviewed, and the standarad criteria applied throughout the study are summarised. By establishing common terminology and notation, this section provides the theoretical lens through which the subsequent design and results are interpreted.

2.1. Motion Control Systems in Laser Micromachining

Laser material processing systems available for industrial production come in a variety of designs, some of which can be classified according to their application. On the one hand, there are general purpose systems that can be used flexibly, and on the other, there are task oriented, specialised systems that are pre-optimised for mass production. Both variants can be parameterised for specific processes, such as laser cutting or drilling [1].

Generally, the basic framework of laser micro-machining systems is always the same. The basic unit is a combination of a power supply, a control unit and a laser head. The laser head consists of an emission unit and a scanner with global focusing optics. Scanners are systems with some movable mirrors that properly steer the laser beam. The base system lacks focus adjustment, a processing platform, and a protective enclosure [2]. These are embedded in a fully developed laser system, which is off the bench and consists of the following elements:

1. Laser base system
2. Housing with beam guidance
3. Axis system
4. Focusing system
5. Control software

This setup allows for a general functional unit schema, as illustrated in Figure 2.1, to be derived for the various system designs.

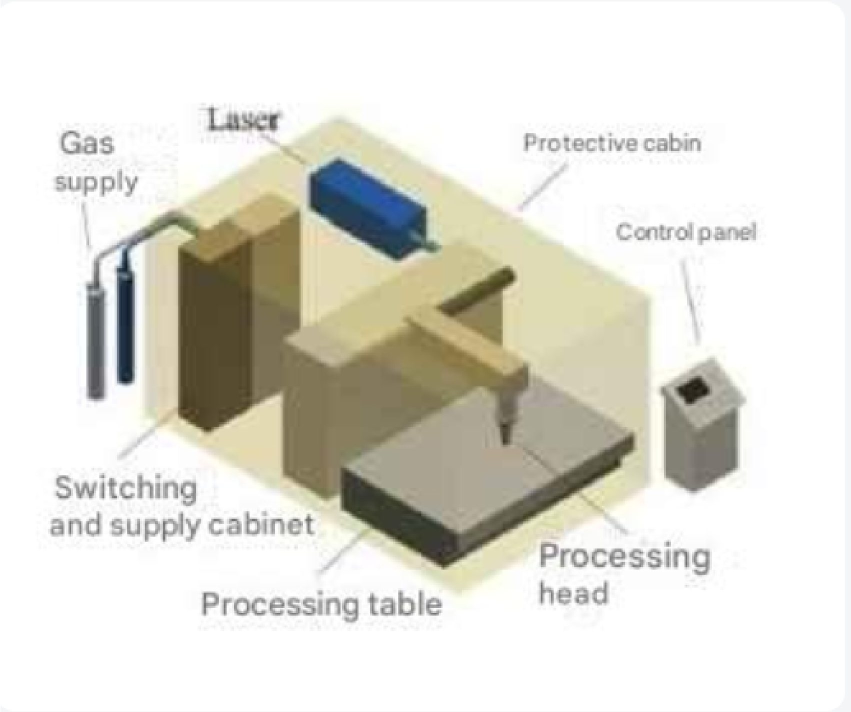


Figure 2.1.: Illustration of the structure of laser processing systems [2].

These components define the quality of the machining results and must be integrated into a production system. Different positioning and transport systems are used for positioning workpieces and tools in the machine workspace, for loading and unloading the machine, and for delivering process components. These systems essentially act as the link between the laser beam and the workpiece. Translational and rotational axis systems are primarily employed. Multi-axis systems enhance flexibility and machining capabilities, though they may reduce achievable positioning and repeatability accuracy unless decoupled from each other [3, 4].

2.2. Structure of Positioning Systems

The positioning system is one of the most critical elements of an automated setup in terms of its functionality and quality. In general, a positioning system consists of a drive inverter with an integrated or external controller, a motor with a position measuring system, a gearbox, and a load driven by the motor. The load can be either a gearbox or a carriage. Drive inverters are categorized as either frequency inverters or servo inverters. Frequency inverters control the motor without measuring the rotational speed, whereas servo inverters provide precise angle and speed control [5]. The control unit acts as an actuator for relative motion between the workpiece and the tool. It calculates the trajectories using an interpolator and provides these as commands to the inverter. The motor then converts these commands into the desired motion [6].

The task of an axis system is to position a workpiece at a defined location or to move it along a desired trajectory to a target point, whether for discrete or continuous machining processes. The focus is on the relative positioning between the tool and the workpiece, either during motion or at a fixed position. The motion sequence and the forces and powers required are key factors. Another task is to transfer mechanical energy to a machine performing a process such as cutting, conveying or grinding. In these cases, the focus is not on spatial movement but on providing mechanical energy for the process. Essential criteria for a drive system include the motion profile and its physical parameters [7].

In ultrashort pulse laser (USP laser) machines, where processes are contactless, positioning tasks take precedence. Since the test routine to be developed focuses on translational drives, only these will be addressed here. Laser machining systems are characterized by high-precision positioning within sub-micrometer ranges and fast travel speeds [8].

2.2.1. Positioning Sensors

Each axis in a positioning system is equipped with a position measurement system to track positional changes. These systems send movement data back to the controller or the position control loop. They can be categorized as direct or indirect, and as absolute or incremental systems.

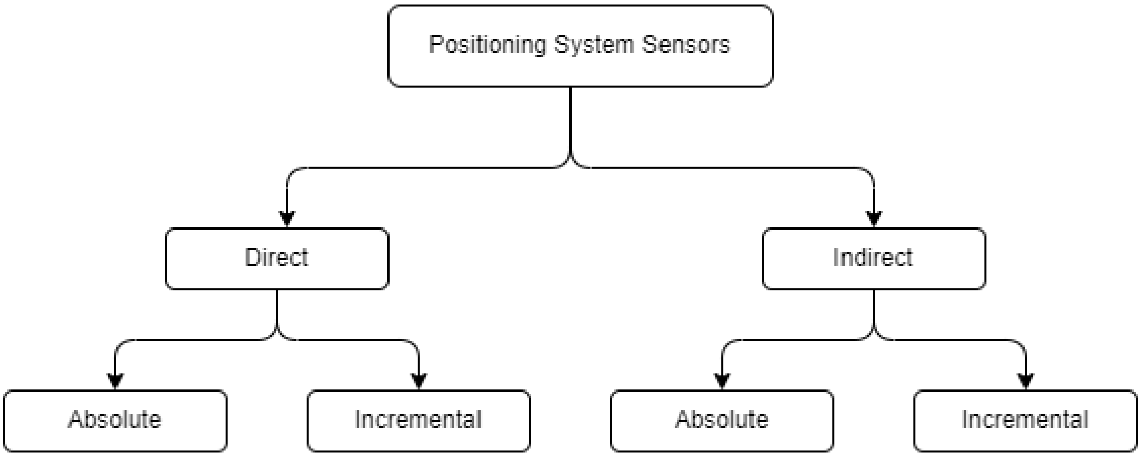


Figure 2.2.: Classification of Position Sensors

In direct measurement systems, positional data is obtained directly from the carriage or moving part of the mechanism. This approach accounts for mechanical inaccuracies, such as backlash or deformation, because these errors are inherently included in the measurement [9, 10]. For example, a scale mounted directly on the moving component might be read by a sensor to determine position.

In contrast, indirect measurement systems collect information from other parts of the mechanism, such as the motor shaft. Such systems make mathematical computations to output a position estimate of the workpiece or moving piece of the machine. However, due to the screw pitch error or the backlash of the transmission system, software compensation needs to be taken care of [9].

Both direct and indirect position measuring systems can be realized incrementally, absolutely or with both variants. The sensors can be digital or analog and are based on different measuring. Analog sensors, like potentiometers, deliver smooth, continuous position information and are, by definition, static sensing systems. In this sensors, a continuous output voltage or current to each position is assigned. Nevertheless, their application range is restricted and for larger measurement range, the analog signal has to be segmented into several intervals.

In the case of digital sensors, data are constantly collected in discrete steps, which makes them ideally suitable for incremental encoding or encoding of absolute values. Such as, the incremental encoders measure the displacement from a reference point using periodic signals, and the absolute encoders assign global digital value to each position using the uniquely coded scale.[11, 12]

2.3. Principles of Dimensional Metrology

Dimensional metrology forms the foundation for precision measurement systems that deals with measuring lengths or distances with high precision, particularly in industrial environments where one-dimensional (1D) measurements are critical [13]. A 1D measurement in precision engineering refers to accurately determining linear displacements or distances along a single axis with high resolution and traceability. Such measurements often require sub-micrometre or nanometre precision over macroscopic travel ranges [14]. Key performance criteria include accuracy, which measures how closely a measured value aligns with its true value, and precision, which describes the consistency of repeated measurements.

In metrology, accuracy reflects systematic errors or offsets, whereas precision is quantified by repeatability the variation observed when the same distance is measured multiple times under identical conditions [15].High repeatability is crucial for many applications, as the ability to reliably produce the same result is often valued more highly than absolute correctness. This is because systematic errors can often be calibrated out, whereas poor repeatability sets a fundamental limit on achievable accuracy. Therefore, a system can be precise but inaccurate if it consistently deviates from the true value, and vice versa [16]. Figure 2.3

provides an example to illustrate this phenomenon

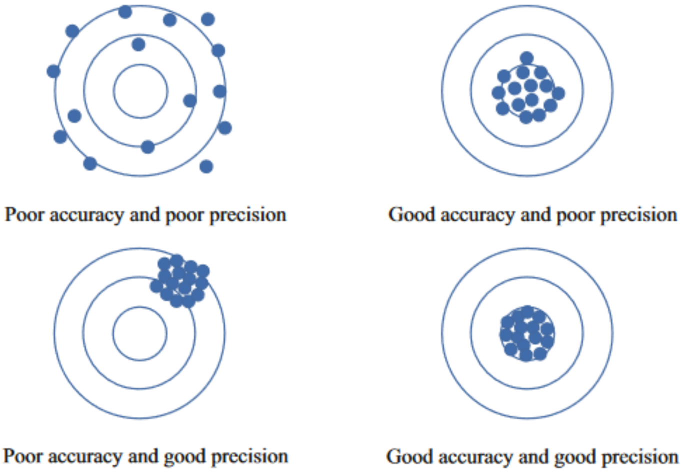

Poor accuracy and poor precision
Good accuracy and poor precision
Poor accuracy and good precision
Good accuracy and good precision

Figure 2.3.: Accuracy vs. Repeatability (Precision)[16].

Achieving accurate 1D measurements requires careful attention to various sources of error, such as thermal drift, backlash, and hysteresis. These factors significantly impact the precision and reliability of measurement systems [17].

Thermal drift happens when temperature changes cause materials to expand or contract. Even a small temperature change can cause expansions or contractions of a machine axis or a scale, leading to inaccuracies over time [18, 17]. Errors such as backlash and mechanical hysteresis have been identified as issues inherent to the mechanics of the motion system. Backlash and mechanical hysteresis are errors associated with the motion system mechanics [18, 19]. Backlash is caused by looseness or gaps between moving components such as between screw threads or gear teeth, such that reversing direction introduces a lag before motion reengage [20]. An illustration shown in figure 2.4.

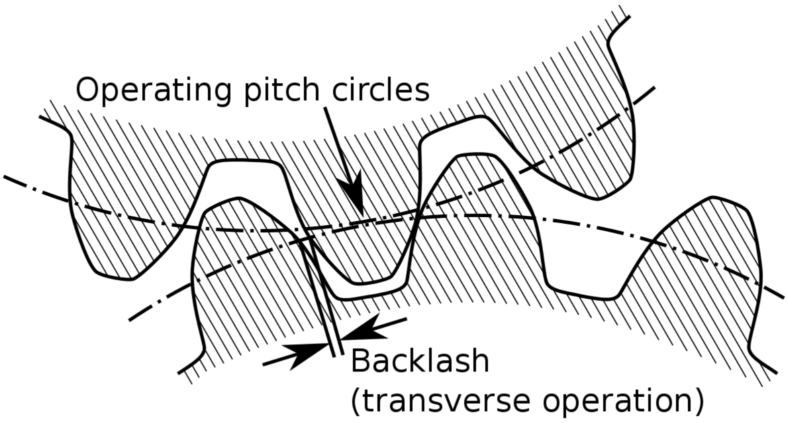


Figure 2.4.: diagram shows how backlash can happen in gear systems[20].

Hysteresis is the elasticity and friction in materials that causes a different return path, the system's output depends on its past inputs, not just the current input, contributing further difference when motion reverses. The net effect is that approaching a target point from opposite directions results in two slightly different positions [21]. This difference is termed the reversal error (or bidirectional error) and is essentially the combination of backlash and hysteresis in the system. In formal terms, reversal error at a given position is defined as the distance between the actual positions reached when the target is approached from the positive versus negative direction [17].

Another crucial concept is positioning deviation, which is simply the difference between the actual position reached and the target position, It represents the fundamental measurement error of the axis at that point. Manufacturers and standards also use the concept of linearity (linear error) to describe how well a system's positioning errors follow a straight-line trend. Linearity error represents the maximum deviation of the measured position from an ideal straight line, typically formed by a best-fit straight line across the measurement range [22].

In summary, high-precision 1D measurement demands careful control of systematic deviation, minimization of backlash/hysteresis (reversal error), and tight thermal and mechanical stability to ensure that measurements are precise as well as accurate.

2.4. Motion Control in Measurement Systems

Achieving precise 1D measurements requires an equally precise motion control system to position the object along the axis. These systems include parts like spindle drives, linear motors and feedback loops, all of which are essential for precise and smooth movement.

In a spindle drives, the rotary motors transmit torque through precision screw mechanisms, translating rotation into linear displacement. While this design offers simplicity and cost efficiency, it introduces potential error sources such as backlash and compliance. Alternatively, direct drive linear motors eliminate mechanical transmission elements altogether, offering higher dynamic response, reduced wear, and improved positioning fidelity [23]. These systems are widely employed in high-speed, high-precision applications like laser micromachining and semiconductor metrology

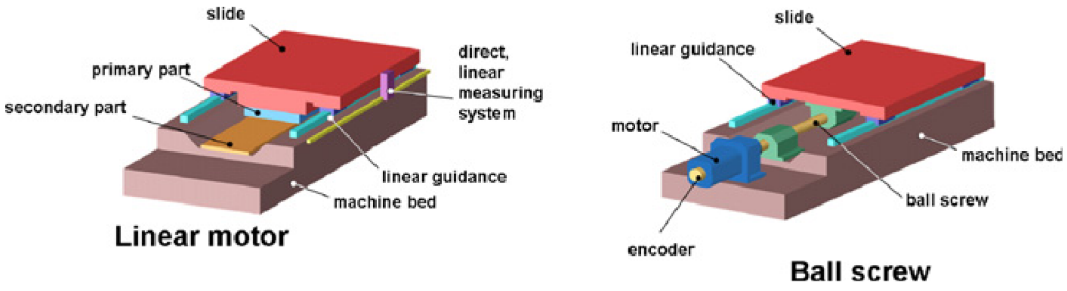


Figure 2.5.: Linear and ball-screw drive mechanisms [23].

Most controllers use a closed-loop feedback system to keep precision positioning systems stable and accurate. This system constantly changes actuator commands in real time to reduce positional errors. Proportional–Integral–Derivative (PID) control is a common method of teaching feedback regulation. However, in the real world, many industrial systems, such as the ACS controller utilised in this study, employ a cascaded architecture with custom control loops. The position loop usually uses proportional (P) control for quick response. The velocity and current loops, on the other hand, use proportional-integral (PI) regulators to keep the error from getting too big and to keep the system stable. Each loop can focus on a different part of motion because of this layered structure. This makes the system more responsive and accurate overall.

A commissioning study at TU Chemnitz showed that identifying a stage's dominant vibration modes during set-up allows default PI/P gains for every cascade loop to be computed analytically, eliminating most trial-and-error tuning and cutting start-up time by more than 30% [TUC1].

The PID controller calculates a control signal aimed at reducing the deference between the commanded position and the actual position reported by a feedback sensor, usually an encoder [24]. This difference is referred to as the control error, denoted as $e(t)$. The controller's output signal, $u(t)$, comprises three distinct terms:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)d\tau + K_d \cdot \frac{d}{dt}e(t) \tag{2.1}$$

- K_p is the **proportional gain**, which scales the present error and provides immediate corrective force.
- K_i is the **integral gain**, which accumulates the error over time to eliminate steady-state bias.
- K_d is the **derivative gain**, which predicts the future trend of the error and adds damping to the system.

This type of feedback control gives the ability to compensate for disturbances and maintain the desired performance. Figure 2.6 shows the structure of a PID controller loop.

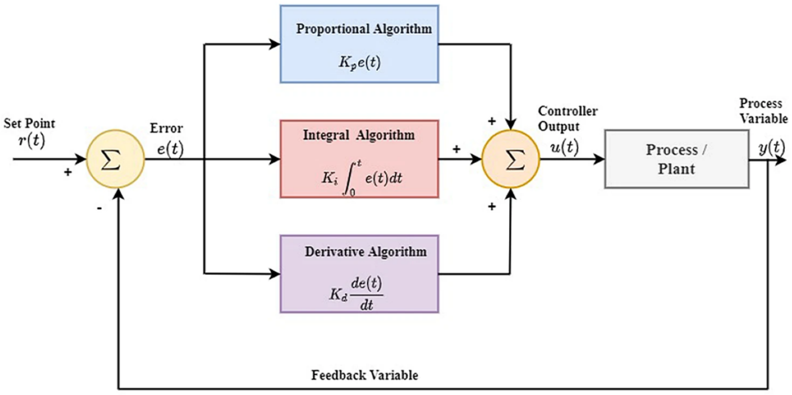


Figure 2.6.: Block diagram of process control using PID [24].

It is important to note that each term of a PID controller plays a unique role in how the system reacts to changes or errors. The proportional term reacts immediately to any error, thereby helping the system make fast corrections. However, if the configuration is excessively aggressive, it can result in the system overshooting the target. The integral term is of pivotal significance in ensuring long-term accuracy by gradually eliminating steady state errors. At the same time, the derivative term is used to help reduce differences by predicting future errors based on how quickly the error is changing [24, 25, 26].

Getting the right balance between these three components often called tuning is crucial. The goal is usually to make the system respond quickly without overshooting or swinging back and forth too much. Ideally, you want a response that's either critically damped or slightly underdamped. Engineers often use step response tests to guide this tuning process [27]. These tests show how the system behaves when asked to move suddenly to a new position. From this, important performance indicators like rise time, overshoot, settling time, and steady state error can be measured and used to evaluate how well the control system is working [28].

In high precision systems, PID control is typically implemented within a cascaded control structure that consists of three nested loops position, velocity, and current control. The innermost current loop regulates the actuator force, the middle velocity loop controls the speed of motion, and the outermost position control loop guarantees precise tracking of the motion trajectory. By allowing each loop to concentrate on a distinct aspect of motion, this layered approach enhances stability and responsiveness, particularly in the event of disturbances or changes in conditions [29]. Figure 2.7 shows an example of a cascaded control loop.

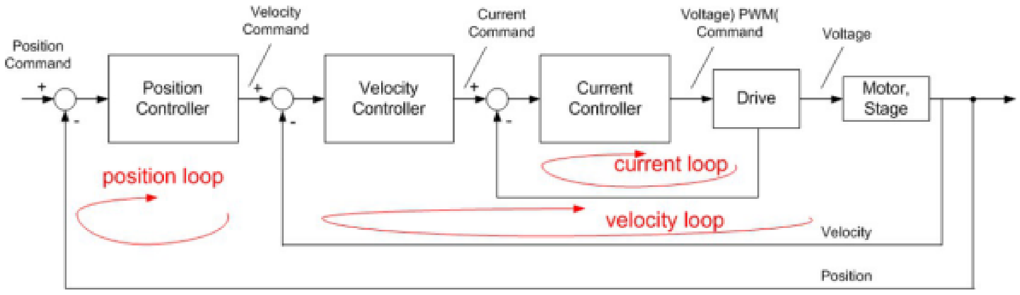


Figure 2.7.: Cascade control for feed controls [30]

In summary, PID control forms the theoretical and practical foundation of precision motion systems. Its ability to adaptively correct motion errors, suppress oscillations, and ensure smooth transitions is central to the performance of any high-accuracy measurement platform. The careful tuning and integration of PID algorithms within digital motion controllers are thus critical for achieving the tight tolerances required in one-dimensional measurement systems.

2.5. Interferometry Fundamentals

One of the foundation of dimensional metrology is optical interferometry, which measures linear displacements with remarkable accuracy. This method takes advantage of light's wave nature, when two coherent light beams are combined, they produce an interference pattern that depends on the difference in their optical path lengths. A change in path length such as when one beam is reflected from a moving stage causes a phase shift between the beams, leading to alternating constructive and destructive interference that can be counted to measure distance [31, 32].

The classic configuration is the Michelson interferometer, which splits a laser beam into two arms using a beam splitter mirror. A fixed reference mirror is reflected off by the reference arm, while a movable mirror attached to the object whose displacement needs to be measured is reflected off by the measurement arm [33]. Upon returning, the two beams recombine at the beam splitter and are directed to a photodetector. As the measurement mirror moves, the optical path length in that arm changes relative to the reference arm, causing the intensity at the detector to oscillate through bright and dark ringes. Each fringe corresponds to a path length change of one wavelength of light (λ) [33, 34] as shown in Figure 2.8.

However, in a Michelson setup, moving the mirror by a distance produces a change of double of that distance in the round trip optical path, since the beam travels to the mirror and back. Thus, one full cycle of intensity (bright to dark to bright) occurs for each half wavelength of mirror displacement. For instance, a He-Ne laser operating at a wavelength of 633 nanometres will generate one interference fringe for every 316.5 nanometres of movement in the mirror. The displacement d can be determined by counting the number of fringes N and using the formula $d = (N \cdot \lambda)/2$ for a Michelson interferometer [35, 36], Figure 2.8 an example of fringe.

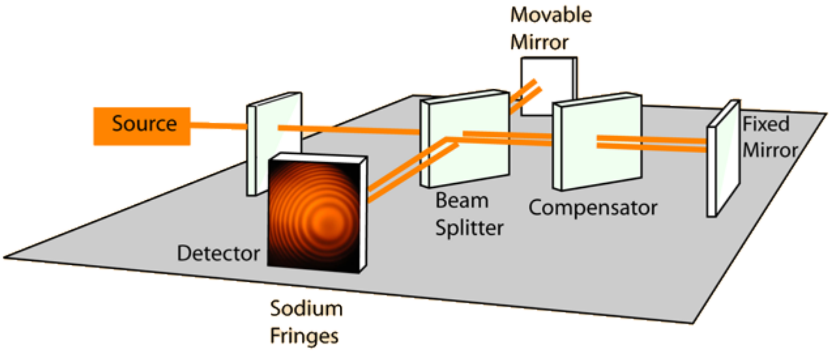


Figure 2.8.: Michelson interferometer setup showing beam paths and resulting interference fringes. Mirror displacement causes fringe shifts proportional to the optical path change[36].

In practice, modern displacement measuring interferometers (DMIs) use stabilized laser sources often helium neon (He-Ne) lasers or diode lasers locked to reference frequencies and optical elements like retroreflectors to ensure alignment and beam parallelism. The use of a single frequency, stable laser with a wellknown wavelength is crucial, because any drift in the wavelength directly translates to a scale error in the measurement [37].

Quadrature detection is commonly used in interferometer systems to enhance resolution and identify motion direction [38]. In order to detect quadrature, two interference signals that are phase shifted by sine and cosine signals must be generated [39, 40]. For example, through polarization optics or phase modulators, the interferometer can output two signals whose Lissajous plot is a circle. The continuous rotation around this circle as fringes evolve provides interpolation of positions between fringe crossings and indicates direction (clockwise vs. counter-clockwise rotation corresponds to motion direction) [39], a figure 2.9 show simple view of the quadrature detection. By electronically subdividing the fringe phase, DMIs achieve resolutions far below a single fringe often to nanometer or even sub-nanometer scale.

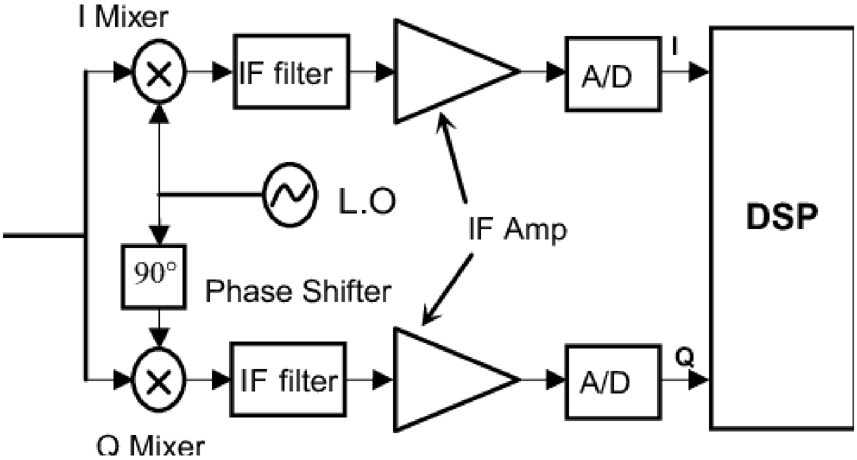


Figure 2.9.: A simple view of the quadrature detection [41].

Interferometry is not limited to linear displacement, multi-axis and multi-channel interferometer configurations exist to measure angular errors and straightness [42]. By sending multiple beams along different points of a moving stage, one can detect pitch, yaw, or straightness deviations. For example, a triple-beam interferometer might send three parallel laser beams, one near the top, one near the bottom and one in between of a moveable mirror or stage [42, 43] as shown in Figure 2.10. If the stage tilts slightly, it introduces an optical path difference between the two beams effectively one beam's path length changes more than the other. By measuring the differential phase between the three channels, the system can infer the angular displacement in addition to linear displacement.

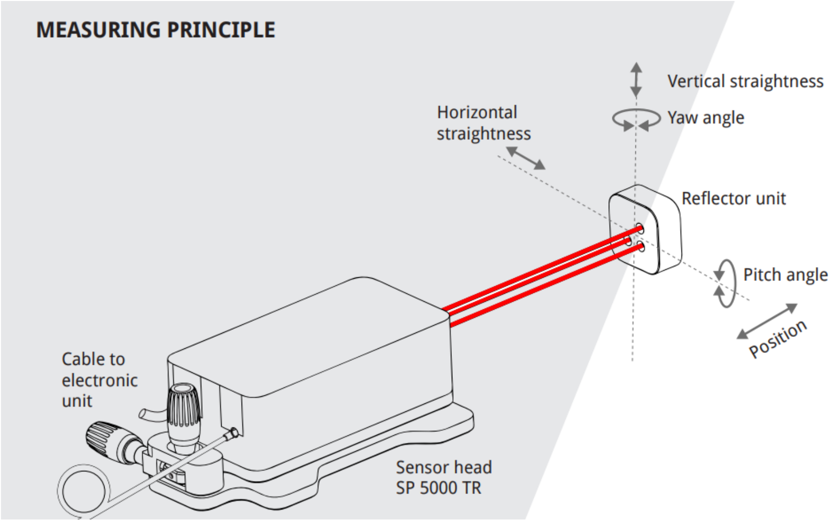


Figure 2.10.: Triple beam interferometer measurement principle [43].

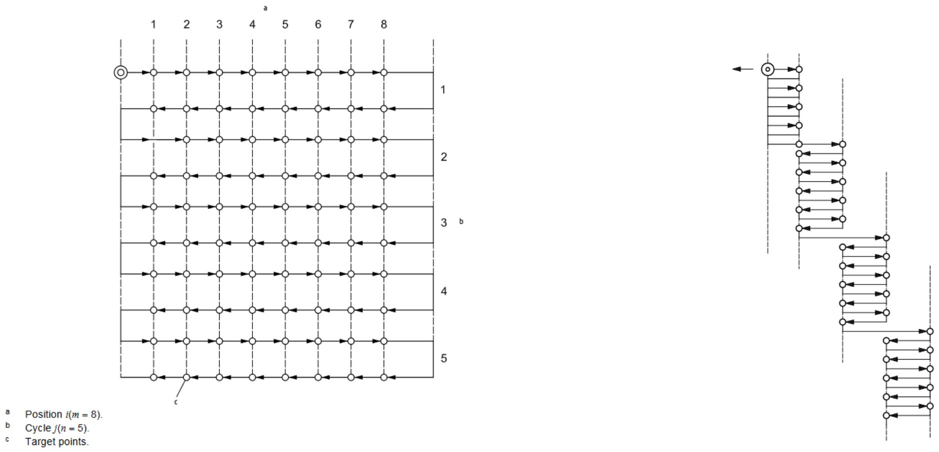
Interferometric measurements are sensitive to environmental conditions and optical setup geometry. A Michelson interferometer is a two beam, unequal path interferometer, meaning each beam travels a different path through air. Thus, environmental influences like temperature, pressure, and humidity can alter the refractive index along one path differently than the other, leading to phase errors. Tiny variations in air temperature or pressure between the two arms induce different phase shifts that can degrade precision [44]. To mitigate this, metrology interferometers often include environmental compensation units that continuously monitor ambient temperature, pressure, and humidity in real time, These values are input into formulas such as the Edlén equation or its updated variants to compute the real-time refractive index of air [44, 45]. This makes it possible for the system to dynamically change the effective laser wavelength and guarantee that the measurement can be linked to its vacuum equivalent [45].

In summary, interferometry provides a robust and highly accurate method for measuring displacement and angular deviation in 1D measurement systems. Its effectiveness hinges on well-designed optical configurations, high-resolution fringe interpolation, and precise environmental compensation. In the context of automated measurement frameworks, these capabilities make interferometers indispensable for validating axis accuracy, detecting motion errors, and ensuring traceability in compliance with international standards.

2.6. Standards for Evaluating Axis Performance

International standards provide a common language for specifying and verifying the positioning capabilities of numerically controlled axes. For linear axis metrology there is two primary standards in this domain which is the ISO 230-2 (International Standards Organization) and the German guideline VDI/DGQ 3441 (a German standard by the Verein Deutscher Ingenieure and Deutsche Gesellschaft für Qualität) [46, 47]. Both describe how to design a test, acquire data and derive quantitative indicators such as positioning accuracy, repeatability, and reversal errors for numerically controlled axes. The terminology introduced in section 2.3 is rooted in these texts. ISO 230-2 is now the globally recognized reference, while VDI/DGQ 3441 remains widely cited because ISO 230-2:1988 was largely modelled on it and later editions retained near-identical procedures [48].

ISO 230-2 (specifically ISO 230-2:2014) is titled "Determination of accuracy and repeatability of positioning of numerically controlled axes." It provides a rigorous framework for testing linear and rotary axes, typically used in machine tools or measuring machines. The standard defines test procedures where the axis is moved to a series of target positions throughout its range. At each target, the position is approached from both the forward direction and the reverse direction, multiple times, and the positions are measured with a reference instrument often a laser interferometer [46] The figure 2.11(a) and 2.11(b) below shows an example of the movement.



(a) Linear motion pattern

(b) Pendulum motion pattern

Figure 2.11.: Standard test cycle

From this data, several parameters are calculated:

- **Reversal error:** Reversal error is the tiny gap that appears when an axis changes direction, revealing how far it overshoots from each side. A perfect mechanism would have none, but real machines see a few micrometers. Measuring the worst and average gaps offers a quick read on backlash, hysteresis, and overall axis health.

- **Unidirectional repeatability:** Unidirectional repeatability gauges how precisely an axis can land on the same spot when every approach comes from the same side. By repeating the move and measuring the spread of hit points, then taking the widest spread seen anywhere along the axis we get its repeatability figure. That number shows how reliably the axis can position itself in one direction moves, free from backlash effects.

- **Bi-directional repeatabilityr:** Bi-directional repeatability measures how much an axis position can drift when it might approach a target from either side. It folds both the random scatter seen in each single-direction and the extra shift caused by backlash into a single spread for each point, then takes the worst spread along the travel. That number captures the true positioning uncertainty making it a key benchmark for machine tool acceptance.

- **Systematic error:** Systematic error, shows how far off an axis can be from the target at its worst. ISO defines this as the range between the largest and smallest average errors across all positions, capturing consistent offset rather than random scatter. It often reflects issues like pitch or calibration errors. A related measure averages forward and backward errors to remove hysteresis effects, it signals uneven behavior between directions.

- **Positioning accuracy:** A positioning accuracy folds an axis's worst systematic offset and random scatter into one 95% (k=2) confidence band. The tighter that band, the better the calibration and consistency.

VDI/DGQ 3441 first appeared in Germany in 1977 under the title "Statistical Testing of the Operational and Positional Accuracy of Machine Tools; Basis" [47]. It addresses the same subjects as ISO 230-2, but with its own terminology and emphasis. The standard helped shape early ISO drafts and is still widely used in Europe for machine acceptance tests [49]. It treats accuracy and repeatability statistically taking the average error, then add a three sigma ($\pm 3\sigma$) band to show the spread, ISO later replaced it with its expanded uncertainty intervals using $k = 2$ for 95% intervals instead of ($\pm 3\sigma$), though the end numbers are similar [47].

In practice, the axis is driven to the same points on multiple occasions, with the actual stopping point being recorded. The mean error and scatter are then calculated. As stated in VDI 3441, it is imperative to plot these errors along the travel trajectory and conduct a reversal test, where the process is cycled back and forth, with the objective of identifying and rectifying any occurrence of backlash, stick slip or servo lag [47]. The system has even been configured to monitor how smoothly the feed runs by logging position error during a steady-speed move [50].

It is important to recognize how crucial ISO 230-2 and VDI 3441 are when it comes to ensuring machine tools meet precision standards. These guidelines help establish consistent methods for certification and calibration, making sure that machine tools are accurate and dependable. Following these standards is essential for both initial acceptance tests and regular checks, ultimately guaranteeing the quality, reliability, and traceability of manufacturing processes.

2.7. Automation and Modularity in Measurement Systems

Modern measurement systems, especially those integrated into industrial environments, benefit greatly from automation and a modular design approach [51]. In this context, automation refers to the use of hardware and software to carry out precise sequences in tasks like data processing, stage movement, and data acquisition with little to no human intervention. A well designed automation system improves not only efficiency and throughput but also consistency of measurements with reducing human error and variation. Modularity is a fundamental principle that complements automation by structuring the system into independent, interchangeable components [52]. In a measurement framework, this might mean separating the motion control module, the sensor interface module, the data analysis module, and the user interface, such that each can be developed and maintained somewhat independently. Each module performs a specific function and communicates with others through well defined interfaces. For example, the motion control module might expose functions like "move to position X" or "home the stage," which the higher level automation sequence can call, without needing to know the low-level details of motor control. In the same way, a data acquisition module could read a laser sensor or gauge, change it to engineering units, and then send the results to an analysis module that does any calculations or corrections that are needed.

Adopting a modular architecture when designing automated test and measurement systems offers several advantages. One of the benefits of scalability is that it allows new functionalities or hardware to be added to the system without having to completely overhaul it. This can be achieved by introducing new modules or upgrading one module.

Another benefit is improved reliability and maintainability. Because modules are independent, a fault in one part is less likely to propagate and crash the whole system; and diagnosing issues is easier since each module can be tested in isolation. If the measurement results seem incorrect, one can focus on the sensor/analysis module; if the motion is behaving oddly, one looks at the motion module, and so on. This isolation of concerns helps in faster troubleshooting and reduces downtime. It also means that upgrades can be implemented by swapping out or updating a module, provided the interfaces remain consistent.

In terms of software engineering, modularity leads to more maintainable and reusable code: functions and classes are organized by component, encouraging code reuse across projects and making the system easier to extend in the future. Indeed, modular software design is a best practice noted in the literature for creating scalable and robust systems. Dr Saleh and colleagues underline this point by showing that lightweight deep-learning models can be rigorously benchmarked for real-time edge deployment, providing a template for performance-critical analytics inside modular automation frameworks [TUC2]

The discussion has clarified the mechanical, control-theoretic, and optical principles that underpin one-dimensional calibration. Functional decomposition of positioning systems, the role of interferometers as traceable length references, and the formal definitions of accuracy (E), bidirectional deviation (A), repeatability (R), and reversal error (B) prescribed by ISO 230-2 now serve as reference points for both the framework architecture and the forthcoming performance evaluation.

2.8. Python in Metrology Automation

Python has become the preferred language for automating dimensional-metrology workflows because it unifies numerical analysis, visual reporting and instrument control within a single, open-source ecosystem. The following subsections outline the technical reasons for this adoption, highlight the most relevant scientific libraries and show how Python bridges the gap between shop-floor hardware and traceable data analysis.

2.8.1. Rationale for Adoption in Measurement Science

Early studies on automated test benches demonstrated that Python's readable syntax and cross-platform binaries reduce development time compared with proprietary scripting environments such as LabVIEW or MATLAB [53]. Subsequent conference work confirmed that the language's low entry barrier encourages maintainable code bases even in safety-critical instrumentation tasks [54]. These findings align with industrial requirements for auditability and life-cycle support in precision-manufacturing software, where standards such as ISO 230-2 mandate transparent calculation chains [46].

2.8.2. Scientific Computing Stack for Metrology

At the core of Python's scientific stack is NumPy, whose contiguous n-dimensional arrays enable vectorised signal processing far faster than interpreted loops [55]. pandas extends this model with labelled DataFrames that simplify the alignment of laser, encoder and environmental channels common in displacement metrology [56]. High-level algorithms for optimisation, interpolation and statistics required by ISO 230-2 are provided by SciPy, whose API maturity has been documented in peer-reviewed literature [57]. Publication-quality error curves and reversal plots are generated via Matplotlib, originally designed to give Python a MATLAB-class graphics engine [58]. For formal uncertainty evaluation, the GUM Tree Calculator automates the propagation rules of the ISO "Guide to the Expression of Uncertainty in Measurement", delivering machine-readable uncertainty budgets that can be embedded into digital calibration certificates [59].

2.8.3. Interfaces to Industrial Measurement Hardware

Python provides superior bindings to the two most prevalent industrial communication layers in metrology, PyVISA is a software that implements the VISA standard, thereby facilitating three-line access to instruments over GPIB, USB, Ethernet or serial buses. This development has the potential to streamline legacy laser-interferometer control [60]. Python-OPCUA is a software that implements both client and server sides of OPC UA. This allows real-time exchange of displacement and environmental data with programmable logic controllers (PLCs) or supervisory control and data acquisition (SCADA) systems without vendor lock-in [61].

Applied libraries such as R-testbench integrate these back-ends with automatic instrument discovery and high-level task orchestration, demonstrating deterministic acquisition at sub-millisecond latency in precision-current metrology. It is evident that, when utilised in conjunction, these tools empower a solitary Python process to facilitate the movement of an axis, the stream of interferometer counts, the computation of ISO metrics, and the publication of results to a manufacturing data lake.

2.8.4. Implications for One-Dimensional Calibration Frameworks

For the linear-axis calibration framework developed later in this thesis, Python's ecosystem yields a coherent data path: NumPy handles raw fringe counts; pandas synchronises temperature, pressure and displacement logs. SciPy computes systematic deviation E, bidirectional accuracy A, repeatability R and reversal error B in accordance with ISO 230-2, and Matplotlib renders diagnostic plots that are embedded into the auto-generated PDF certificate. Hardware control is orchestrated by the existing VBScript wrappers inside microSTAGE, while the Python layer focuses exclusively on data reduction and visual reporting. This separation keeps the metrological computation transparent and version-controlled without altering the proven motion-control stack.

3. State of the Art

Before proposing a new framework, it is essential to position current industrial practice and academic research in automated one-dimensional measurement. The text first documents the fragmented workflow presently used at 3D-Micromac spanning discrete motion-control interfaces, interferometer utilities, and spreadsheet post-processing and then canvasses alternative commercial and research solutions. The comparison highlights shortcomings in cycle-time efficiency, data traceability, and standards compliance, thereby motivating the design choices adopted later.

3.1. Current Workflow at 3D-Micromac

An essential element of 3D-Micromac's production environment is microMMI an internally developed tool-chain platform written in C++. microMMI acts simultaneously as the machine-control hub and the operator user interface. It orchestrates communication with motion controllers, safety interlocks, vision modules, and peripheral sensors, while exposing VBScript hooks that allow process engineers to automate axis movements, laser parameters, and auxiliary routines in a single workspace.

Nested inside this platform is microSTAGE, a specialised application that is currently limited to 2D measurement and calibration. Using a calibrated wafer and a high-accuracy camera. Although it leverages microMMI's communication backbone, it does not yet handle 1D laser interferometric axis calibration.

At 3D-Micromac, calibrating the positioning axes is an important step in making sure that machines are accurate and that quality control is done before systems are sent to customers. Although Laser interferometry gives the necessary resolution and traceability for measuring axes, but the current method has a fragmented and mostly manual workflow, which makes things less efficient and makes it harder to repeat.

The workflow is centered around a collection of linked but separate tools. Commands for linear motion are carried out via a specific axis control interface, usually an industrial-grade controller like ACS motion [62]. To measure displacement, an external interferometric system, like SIOS [63], is used to record the raw position data while the axis is moving. Unfortunately, these systems tend to function as independent applications, with minimal integration between the control mechanisms and the measurement processes.

The transfer of data between these tools occurs manually. Logs of measurements generated by the interferometer system must be exported and subsequently processed externally, often using software like Microsoft Excel. The metrics established by ISO 230-2 such as positioning deviation (E), repeatability (R), and reversal error (A) are computed through user designed spreadsheets using Excel formulas. This manual post processing introduces significant risks, such as inconsistent formula application, version discrepancies, and the potential for errors during copy-pasting. Furthermore, any adjustment to calibration parameters necessitates a complete restart of the measurement and evaluation cycle, hindering the ability to efficiently engage in iterative tuning or comparative diagnostics.

The processes of reporting and protocol generation also follow an equally disjointed method. Final results are compiled manually and lack a standardized format, complicating traceability, particularly during audits or when formal documentation of machine acceptance is required. Consequently, despite the high precision of the interferometric equipment, the overall workflow remains labor-intensive and excessively reliant on the diligence of users.

This traditional methodology also poses challenges to scalability. Each segment of the workflow ranging from the execution of motion commands to the final reporting stage is tied to a distinct software environment. Consequently, any adjustments for new axis configurations, machine types, or evaluation criteria demand significant manual input or modifications to existing scripts. Furthermore, considerations for environmental factors such as temperature, pressure, and humidity are only partially integrated and are not dynamically accounted for during the measurement process, further undermining the accuracy of results under varying ambient conditions.

In light of these limitations, the necessity for an integrated and automated measurement framework becomes clear. The current workflow's dependence on manual coordination between disparate tools, the absence of real-time feedback, and the inconsistencies in documentation all contribute to avoidable errors and delays. These shortcomings underscore the imperative for developing a cohesive, script-driven calibration environment, which will be elaborated upon in the subsequent chapters of this work.

3.2. Calibration Techniques for Automated 1D-Measurement

Traceable 1-D displacement metrology historically relied on gauge-block interferometry; however, industrial adoption of 24-hour production cycles and the demand for sub-micrometre uncertainties have shifted the focus toward fully automated laser-based procedures that satisfy ISO 230-2 acceptance and VDI 3441 monitoring. The following review summarises the academic evidence supporting that evolution and explains why heterodyne and self-tracking interferometers dominate contemporary calibration workflows.

Gauge-block interferometry as the primary standard. Ikonen et al. built an automated interferometer that used a single stabilised He–Ne source and a white-light source to realise the metre with deviations below 10 nm for gauge blocks up to 100 mm, thereby establishing the uncertainty floor against which all secondary methods are compared [64]. Although the technique remains indispensable for national laboratories, its manual wringing procedure and stringent environmental requirements render it impractical for shop-floor automation. An example of a gauge block is shown in the figure 3.1.

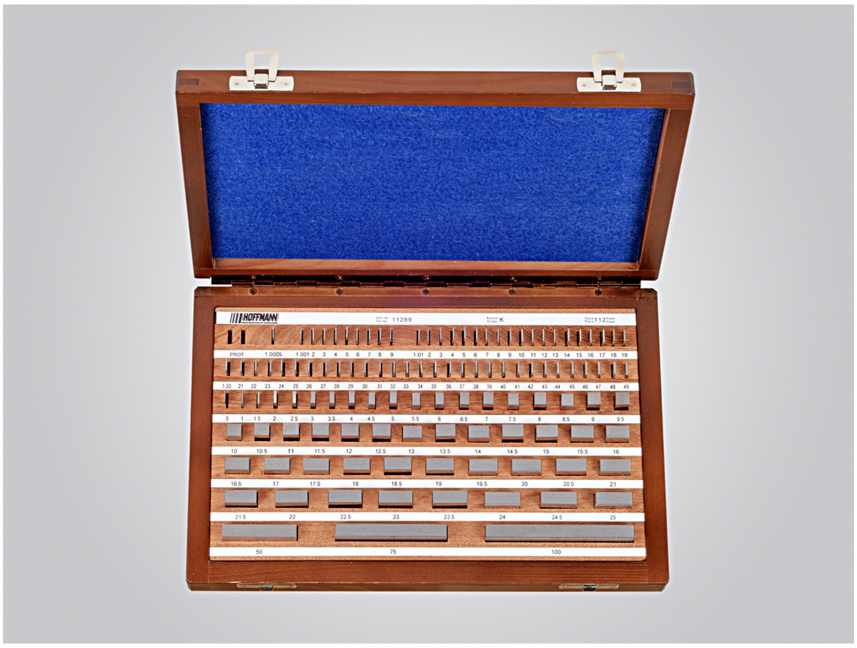


Figure 3.1.: Metric gauge block set

Axis-level calibration under ISO 230-2. Lee et al. demonstrated that a self-tracking LaserTRACER executing the complete ISO 230-2/-6 test sequence can map positioning errors below ±0.5 µm over 1 m in a single set-up, reducing test time by 70% relative to conventional static interferometers [65]. Comparative studies show that positioning accuracy, backlash and bidirectional repeatability obtained via VDI 3441 correlate with the ISO metrics to within 3%, confirming methodological equivalence when identical data treatments are applied [66].

System-level calibration of heterodyne interferometers. Haitjema reviewed every uncertainty contributor laser frequency, counting electronics, optics and environmental probes and recommended recalibration intervals anchored to primary standards, noting that Zeeman-stabilised frequency drift, rather than cyclic non-linearity, dominates long-range error budgets [67]. Berkovic and Shafir independently corroborated that conclusion in a broader survey of optical displacement methods [68]

Environmental compensation. Air refractive-index variation contributes up to 1 $\mu m\ m^{-1}$ under typical factory climates. Brosed et al. integrated a Ciddor-based regression model inside a tracking interferometer and reported residual refractive-index errors below 2×10^{-1} over an eight-hour machining cycle, enabling volumetric verification without beam-specific sensors [24].

Multi-degree-of-freedom error separation. Differential architectures now target angular misalignment and Abbe offsets that escape classical cosine-error models. Yan et al. eliminated rotational crosstalk below 20 nm by means of a differential phase-modulated design employing a 2 × 2 avalanche photodiode array [69]. Complementarily, Tran et al. extended ISO acceptance to rotary axes by multilaterating a corner-cube trajectory the so-called Laser R-Test thereby suppressing tilt-to-length coupling by an order of magnitude and achieving angular uncertainties of 2 arc-sec [70].

Sequential multilateration and automated alignment. Ezedine et al. showed that design-of-experiments optimisation of sequential multilateration parameters reduces machine-tool calibration time by 40% without degrading volumetric accuracy [71]. Yu et al. integrated an auto-alignment platform inside a Fabry–Pérot sensor head, achieving ±0.05 µrad beam-steering accuracy and completing an entire linear-axis calibration in 75 s, fully unattended [72].

Traceability in on-machine measurement. Despite these advances, Muelaner and Calleja caution that on-machine probing remains vulnerable to thermal drift unless closed-loop environmental data are logged and linked to the digital calibration certificate, a prerequisite for true traceability [73].

The accumulated literature therefore converges on three imperatives that underpin next-generation 1-D calibration: continuous monitoring of laser frequency to mitigate ageing and mode hops; real-time environmental sensing with sub-0.01 °C and 0.01 hPa resolution; and multi-degree-of-freedom error separation through differential or multilateration optics. When implemented collectively, these measures reduce expanded uncertainties to below ±0.25 µm over 500 mm of travel an order of magnitude improvement on manual gauge block routes and a decisive enabler for fully autonomous calibration cells.

3.3. Motion Subsystems for High-Precision 1D Measurement

Precise 1D measurement depends as much on the mechanical quality of the motion subsystem as on the sensor that closes the metrology loop. Positioning stages must combine long travel, nanometre-scale repeatability and high dynamic bandwidth while remaining compatible with interferometric feedback. Contemporary research therefore pursues three complementary design lines, ball-screw servo drives refined by model based error compensation, direct-drive linear motors supported by friction-free bearings, and piezo-actuated flexure mechanisms that supply high bandwidth nanometre correction.

Servo-driven ball-screw stages. Modern ball-screw feeds remain attractive for metre-scale travel because of favorable stiffness to cost ratios. Liu et al. embedded a cascade-forward neural network inside the feed-forward channel of a CNC axis and reduced the maximum tracking error by 77% relative to classical transfer-function compensation [74]. Complementary metrological work shows that lead-error indicators that were 6–40 µm before compensation can be reduced in situ to roughly 6 µm after combined geometric-tilt and eccentricity compensation [75]. Although reducer elasticity and mechanical play still limit speed-loop accuracy, adding a dual-encoder scheme reduces the reducer-side speed fluctuation from about ±10% to ±4%, yielding roughly a 2 ½ fold improvement in low frequency vibration behavior [76].

Direct-drive linear motors on non-contact bearings. Ironless permanent-magnet linear synchronous motors (PMLSM) eliminate mechanical transmission errors and backlash. The end effect analysis of an ironless slot less long stator PMLSM running on an air bearing shows that thrust ripple rate can be cut from 1.9% to about 0.4% of the average thrust by lengthening the mover's end iron, without resorting to Halbach magnet arrays [77], whereas Yu et al. employed a differential dual-linear-motor drive to suppress stick–slip at very low speeds, reducing the critical creeping velocity to about 1 mm s^{-1} and achieving stable micro-feed operation below that threshold [78]. These motors reach their full potential only when coupled to friction-free guidance. Zhao et al. catalogue recent strategies for boosting the load capacity and stiffness of aerostatic thrust bearings, highlighting micro-orifice arrays, annular compensation and porous restrictors. While the review does not fix universal performance targets, it notes that state-of-the-art ultra-precision spindles have axial stiffness well above $100N\mu m^{-1}$ and that parametric studies continue to push pad-level load capacity and stiffness higher [79]. A flexure assisted air-bearing XY stage showed a 200 mm travel with maximum yaw-motion error limited to about 0.9 arc-sec and static yaw stability of 0.008 arc-sec, demonstrating that flexure-based yaw compensation can keep Abbe errors at the sub-micrometre level over long strokes [80].

Control of electromagnetic and cogging disturbances. Slot-less permanent-magnet linear motors are routinely chosen for precision stages because the absence of stator teeth almost eliminates cogging force, though at the cost of 10-20% lower continuous thrust density. Reviews of cogging-force-mitigation techniques confirm that properly designed slot-less or fractional slot layouts can push residual cogging below 0.5% of rated thrust, sometimes to 0.1% with auxiliary skewing or end-effect optimization [81, 82]. Force-ripple modelling of a square-coil, Lorentz-force planar motor likewise reports near-zero cogging and ripple after coil-geometry optimization [83].In a dual-Halbach tubular actuator, embedding thermal limits in the optimisation loop kept the winding temperature at 80 °C while delivering a simulated $2.6 - 2.9 \times 10^5 Nm^{-3}$ volumetric force density ($\approx 30 - 38$ Ncm^{-2} for the prototype length) [84]

Piezo-actuated flexure nanopositioners. Ball-screw or linear-motor coarse stages are increasingly paired with piezo-flexure fine stages to close the nano-scale gap. A 2024 review of compliant piezo devices highlights efforts to reach > 100 μm stroke at kHz-scale resonance, with closed-loop repeatability approaching the nanometre range [85]. A quad-parallel flexure stage with high-stiffness PZT drive demonstrated 53 μm stroke and 1315 Hz natural frequency, enabling high-speed fine positioning [86]. Adaptive sliding-mode control using strain-gauge feedback further reduced residual hysteresis to below 3 nm peak-to-peak during 50 Hz triangular tracking [87].

Integration outlook. Comparative benchmarking indicates that precision ground ball-screw stages equipped with dual encoders can achieve expanded (k = 2) positioning uncertainties in the $0.2 - 0.4$ μm range over 500 mm travel at a parts and assembly cost roughly 40% lower than equivalent direct-drive solutions. Ironless linear-motor stages mounted on aerostatic or hydrostatic bearings routinely deliver repeatability below 20 nm and straightness better than 0.3 μm over 300mm in controlled environments, albeit at the expense of higher energy consumption and more demanding thermal management.

Piezo-flexure modules, meanwhile, offer sub-nanometre step resolution and closed-loop repeatability of 2–5 nm, making them ideal for tip-in or dithering corrections rather than full-stroke motion. Consequently, state-of-the-art

automated 1-D metrology axes increasingly adopt a macro–micro architecture, a linear-motor air-bearing coarse stage provides long-range travel and passive vibration isolation, while an embedded piezo-flexure nest handles high bandwidth nanometre-level error suppression.
Operated under dual-loop interferometric feedback and data-driven error mapping, recent demonstrations have achieved bidirectional repeatability of $\pm 0.10\ \mu m$ for short-stroke ($\leq\ 150mm$) linear-motor air-bearing stages using PI V-480 actuators[88] and $\pm 0.25\ \mu m$ over 500mm travel for macro–micro systems that pair an Aerotech PRO190LM coarse axis with a flexure-guided PZT nest [89], illustrating the range–accuracy trade-off that shapes current axis-design research.

3.4. Laser Interferometry Sensors

For the proposed automation cell only two interferometric families are essential: a fast heterodyne channel that closes the servo loop and a slow absolute-reference channel (synthetic-wavelength or dual-comb) that periodically verifies scale factor and eliminates cumulative drift. Other architectures such as frequency-scanning, fibre Fabry–Pérot or self-mixing probes are therefore mentioned only briefly as future options.

A commercial Zeeman-stabilised dual-frequency He–Ne head (e.g., Renishaw XL-80) specifies a linear measurement accuracy of ±0.5 ppm over the full 80 m range, while the internal laser frequency drifts by less than ±0.05 ppm over three years [90]. A heterodyne implementation that uses a 358 MHz acousto-optic frequency shift has been demonstrated for HIFU (High-Intensity Focused Ultrasound) pressure metrology. The authors achieved a $\sim\ 150\ MHz$ measurement bandwidth around the carrier and resolved 40 MPa peak-to-peak pressures, although the displacement noise density and high-frequency SNR were not quantified [91]. Periodic non-linearity remains the dominant residual, typically 2–5 nm peak-to-peak [92].

A phase-locked synthetic-wavelength interferometer that steers an external-cavity AOM suppresses cyclic error to $<$ 0.05 rad and, with a 0.682 m synthetic wavelength, shows an Allan deviation of $\approx 32\ \mu m$ at 1 s averaging. The authors demonstrated stable locking while stepping a 480 mm path and confirmed that no dead-zone occurs beyond $\approx\ 6\ m$ of free-space range [93]. Where range must extend far beyond the machine envelope, dispersive Fourier-transform dual-comb lidar achieves 262 nm single-shot precision and 2.8 nm after 1.5 ms averaging across 1.7 km numbers far exceeding shop-floor needs but confirming that comb referencing decouples accuracy from distance [94].

Frequency-scanning interferometry can recover absolute lengths of 70–1400 mm to within 1 µm after quadratic refractive-index compensation [95], offering a low-cost alternative if synthetic-wavelength components are unavailable. Fibre Fabry–Pérot probes and self-mixing sensors provide sub-nanometre resolution in miniature vacuum compatible heads [95]. but are excluded from the present design because the axis already accommodates free space beam paths.

In summary, adopting a dual-channel scheme heterodyne for real-time feedback, synthetic-wavelength (or comb) for periodic self-calibration meets the framework's accuracy target of $\leq \pm 0.25\ \mu m$ over 500 mm travel while keeping hardware complexity and integration risk low. Additional sensor types remain valuable for future upgrades but are not required for the baseline implementation.

The review confirms that no single off-the-shelf platform simultaneously addresses deterministic motion orchestration, in-line ISO analysis, and automated certificate generation. Commercial packages such as Renishaw CalConfig focus on metrology but assume controller integration, whereas generic automation suites lack metrological depth. The identified gaps particularly manual data hand-offs and non-standardised calculations directly inform the modular, script driven concept developed subsequently.

4. Concept

The shortcomings identified earlier are translated here into concrete design objectives, yielding a modular concept for an integrated calibration framework. Five interacting modules Motion Control, Measurement, Analysis, Man–Machine Interface, and Reporting are connected through file-based APIs and scripted calls. Particular emphasis is placed on data provenance, scalability, and the separation of deterministic computation from user interaction.

4.1. Design Overview

The automation framework developed for this thesis brings the entire 1D measurement workflow into the microSTAGE platform, replacing what used to be a series of disconnected manual steps with a single, unified automated process. This solution bring motion control, data collection, analysis, calibration, and reporting all together in one place so the user can manage everything from a single interface.

At the heart of the system is microMMI, which serves as both the control center and the user interface. From setting up the measurement parameters to viewing final results, every step happens within this environment. The system is built using a modular approach each major function is handled by its own dedicated component. That includes moving the axis, collecting data from the interferometer, running the analysis with an external Python script, and generating reports through an internal engine.

Motion control is handled by an ACS controller, selected for its compatibility with the existing setup and offers a solid, robust API. All movement commands are sent directly from microMMI, giving full control over axis parameters such as velocity, acceleration, and jerk. Once the axis reaches the desired position, microSTAGE pulls in the latest data from the INFAS software via its API. This includes both positional readings from the interferometer and environmental factors like temperature and air pressure. The system repeats this process across the range and number of points specified by the user.

Once all the data is collected, microSTAGE is handed off to an external Python-based analysis engine. Python was selected for this role due to its extensive ecosystem of scientific libraries, ease of integrating with CSV data, and its proven use in rapid prototyping and numerical analysis. Even though this part runs separately from the main automation loop, it's still tightly integrated. The platform keeps track of the script's progress and handles the output automatically. The script performs statistical evaluation based on ISO 230-2 to calculate positioning accuracy, repeatability, and systematic errors. with results returned in a structured JSON format.

The final report is based on that JSON file, which is exported as a PDF using a fixed HTML/JavaScript template within microSTAGE. This ensures consistent formatting and keeps the reporting process fully integrated into the system.

The design prioritizes traceability, user accessibility, and system modularity. Each component is loosely coupled but connected through well-defined interfaces. Figure 4.1 and 4.2 (or in Appendix A) illustrate the shift from the current fragmented process to the integrated architecture developed in this thesis.

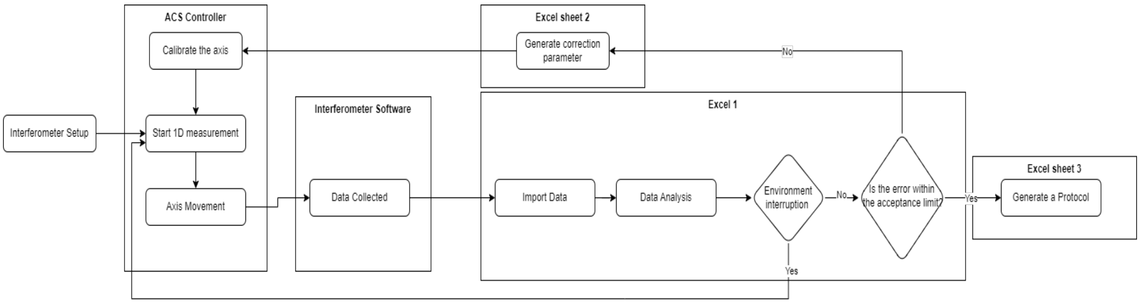


Figure 4.1.: Current Workflow

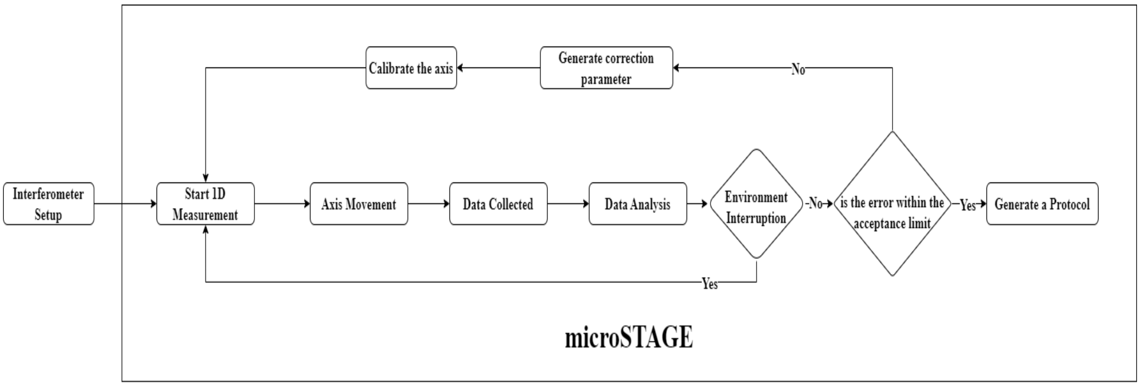


Figure 4.2.: Expected Workflow

This design allows 3D-Micromac to move from a tool chain dependent measurement process to a unified, modular, and scalable platform embedded within its existing automation infrastructure. The system design also enforces synchronization between components, minimizes manual intervention, and enables standardized reporting in compliance.

4.2. System Components

The following sections describe each of the system's core components, outlining their role, internal logic, and integration within the overall framework.

4.2.1. Motion Control Module

The motion control module handles all the linear and rotary movements needed during the 1D measurement process. It's fully built into the microSTAGE software environment, which means users can configure and run motion routines directly through the system's user interface. This makes the process smooth and consistent with how the machine is already used, without needing any external tools or controllers.

For this project, the system was set up to work with an ACS motion controller a powerful and widely-used device at 3D-Micromac. The communication between microSTAGE and the controller happens through its own API, which allows direct control over basic functions like finding the home position, moving to specific points, making relative moves, and checking real-time feedback. All these commands are triggered from within microSTAGE scripts, making it easy to coordinate motion with other tasks like data collection and analysis.

During a normal measurement session, the axis starts by going through a homing process to find a fixed reference point. Then, based on settings provided by the user like where to start and end, how many points to measure, how far each step should be, and how many times to repeat. the system creates a movement plan. This plan is carried out in a loop that moves the axis to each specified location.
After every move, the system waits a moment to make sure everything has settled completely before starting the next measurement. This helps prevent errors caused by vibrations or incomplete movement.

To maintain safety and precision, the system incorporates software-imposed motion limits, which are obtained from the controller at startup and verified against user inputs. If a user attempts to define a measurement range beyond these limits, microSTAGE will prevent execution and prompt for adjustments. These safeguards significantly reduce the likelihood of hardware collisions or unintended out-of-range movements.

The system also keeps an eye on itself while running if there's a problem with the motion or if communication with the controller drops, it will catch the error, and halts the system execution and logs the issue for traceability.

Because the motion control logic is built in a modular way, it's easy to expand in the future. Whether adding more axes, or even working with angular positioning, the same framework can be extended to support new use cases. This ensures that the microSTAGE platform remains flexible and scalable for years to come.

4.2.2. Measurement Module

The Measurement Module is designed to accurately capture high-precision distance and environmental data in perfect sync with axis positioning during 1D measurement processes. It connects directly with the INFAS interferometer software, which continuously records internal sensor data using its own high-speed acquisition system.

In the architecture developed, the INFAS software logs measurement data in real time. The microSTAGE platform accesses this data through the INFAS-provided API. When the motion control module confirms the axis has stabilized at the desired position, microSTAGE sends a data retrieval command to the interferometer API. At this moment, the most recent interferometric measurements and environmental parameters are fetched for immediate use.

In the implemented system architecture it ensures tight coordination between data acquisition and axis movement controlled by microSTAGE. Once the axis reaches a predefined position and motion has fully settled (as confirmed by the Motion Control Module), microSTAGE sends a query to the INFAS software via its dedicated API. microSTAGE captures the latest value from all configured channels in a defined sample time.

For each measurement position, key parameters extracted include linear displacement measurements from three interferometric channels, along with environmental variables such as temperature readings from three sensors, atmospheric pressure, and the operative laser wavelength. These parameters provide critical context for subsequent analysis, enabling adjustments for thermal drift, pressure changes, and wavelength fluctuations that could affect measurement accuracy.

To improve data reliability and reduce random noise, the system collects multiple samples at each measurement pointThe number of samples usually ranges from 5 to 20, depending on the specifications of the interferometer. The final stored values for each channel and parameter represent the average of these measurements, significantly enhancing stability and accuracy by smoothing out transient fluctuations or sensor noise.

The complete buffer of measurement data is then written to a structured internal data file (such as CSV or JSON format). Each line in the file corresponds to a single axis position and includes all measured and derived parameters. This file is saved at the end of the measurement cycle and passed to the Analysis Module, a Python-based external processing unit. This separation of acquisition and processing allows the same raw measurement set to be reused multiple times for:

- Alternate calibration routines
- Debugging or comparison
- Post-run data visualization or report regeneration

This design also permits retrospective analysis if measurement anomalies are discovered, eliminating the need to repeat the test physically.

All data is timestamped, position-indexed, and verified during acquisition. Any communication or readout errors from the interferometer API are logged, and the system retries acquisition within a specified timeout period before halting. This ensures resilience against temporary API or sensor failures.

By gathering a rich dataset that includes environmental factors alongside primary position data and organizing it in a reusable format, the Measurement Module acts as a dependable bridge between motion execution and high-level analysis. Its architecture facilitates modular development, supports traceability, and meets the precision standards required by relevant industry regulations.

4.2.3. Analysis Module

The Analysis Module is built to handle measurement data with enough flexibility to work with different standards. For this thesis, all calculations and evaluations follow the ISO 230-2 standard a widely accepted method for assessing the accuracy and repeatability of CNC machine positioning axes. The module itself is written in Python and runs separately from the real-time data collection process.

Once a measurement session is done, microSTAGE saves the interferometric and environmental data into an organized CSV file. This file includes raw data from both the measurement channels and environmental sensors. Then, microSTAGE triggers a Python script via a shell command to begin the analysis. A VBScript monitors and log this process to make sure everything runs smoothly. During execution, the system keeps an eye out for any errors or interruptions, ensuring that the external analysis completes safely.

The analysis implementation for this project complies with ISO 230-2 standards. It performs calculations including position deviations at various points, the repeatability of results across multiple runs, and the systematic errors between forward and backward movements. These calculations help identify patterns such as consistent offsets and provide insight into how the axis behaves in both directions.

The analysis results are consolidated in a summary table of key ISO 230-2 evaluation parameters, as shown in Table 4.1. This table presents essential metrics such as positional deviation (E), bidirectional positioning deviation (EB), repeatability (R), and reversal error (DR). Each parameter is derived from the raw measurement data and computed in accordance with the mathematical definitions and procedures outlined in the ISO 230-2 standard, ensuring accuracy and comparability.

Table 4.1.: ISO 230-2 Evaluation Metrics Used in the Analysis

Metric	Symbol	Description
Bi-directional accuracy	A	Max positioning error from both directions.
Unidirectional accuracy	A↑, A↓	Max error from one direction only.
Bi-directional systematic deviation	E	Average deviation from setpoints in both directions.
Unidirectional systematic deviation	E↑, E↓	Average deviation when approached in one direction.
Range of mean deviation	M	Range between max and min average deviations.
Bi-directional repeatability	R	Spread in repeated positioning from both directions.
Unidirectional repeatability	R↑, R↓	Spread in repeated positioning in one direction.
Reversal value	B	Difference when reaching a point from opposite directions.
Mean reversal value	$\bar{B}$	Average of all reversal values.

After processing, all findings are saved in a detailed JSON file. This file includes not only the main metrics but also statistical summaries and optional calibration data. Alongside the numbers, visual outputs like deviation curves, error plots, and repeatability graphs are created. These visuals help users understand how the machine axis performs across its full range and point out where adjustments might be needed.

Once the analysis concludes, the data is imported back into microSTAGE. Operators review the metrics and plots within the system's MMI. At this point, the user can choose to store the generated calibration parameters into the system for future compensation or proceed to generate a formal measurement protocol. This flexibility allows the same measurement data to be reused for different evaluations without repeating the physical process.

By keeping the analysis separate from the real-time control system, the design ensures that the main system remains responsive and stable. It also creates a clear, reusable workflow for processing data, which supports traceability and makes it easier to meet metrology standards like ISO 230-2.

4.2.4. Man-Machine Interface (MMI)

The Man-Machine Interface (MMI) is the part of the system that users interact with directly. It forms the front end of the 1D measurement automation setup and is fully embedded within the microSTAGE environment. Since microSTAGE is built on the microMMI platform which serves as the main user interface framework for all 3D-Micromac systems the Measurement Module benefits from a consistent look, feel, and functionality across different features and applications.

Through the MMI, operators can set up all necessary parameters before starting a measurement cycle. This includes choosing the type of movement either a linear motion or a pendulum style pattern. Users also define the range of the measurement by setting start and end positions. Additional settings include how many points will be measured, how many times the full movement should repeat, and optionally, whether to apply a small random offset between repetitions.

In addition to motion settings, the interface gives users full control over axis behavior. They can adjust parameters like velocity, acceleration, jerk of movement, and wait time between steps. This level of customization makes it possible to fine tune the system for different setups, sensor types, or precision requirements.

During measurement execution, the MMI provides real-time feedback to the operator. Status logs are continuously updated to reflect the system state, error messages, and execution milestones. Live values from both the axis controller and interferometer channels are displayed to allow visual monitoring of system behavior. This immediate feedback loop enhances usability and helps operators detect anomalies during execution without halting the cycle.

During measurement execution, the MMI keeps the operator informed with real time updates. Status logs are continuously updated to reflect the current status of the measurement, any errors that occur, and key progress markers. At the same time, live data from both the motion controller and interferometer sensors are displayed. This lets users visually track how everything is performing as the test happens, improving transparency and helping catch issues early without interrupting the process.

The complete measurement process as managed through the MMI is outlined in Figure 4.3.

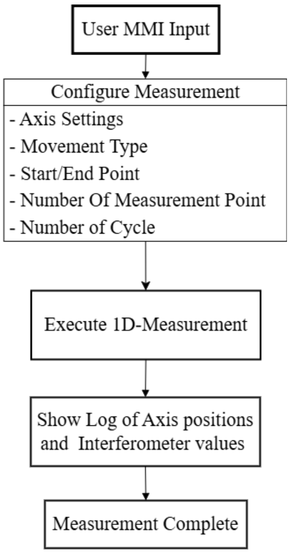


Figure 4.3.: User input and measurement execution flow in the MMI interface.

Upon completion of the measurement cycle, the system moves into the results review phase. The data is presented in structured tables and visual graphs, allowing immediate interpretation of the findings. These results are saved locally on the system, making them available for later reference or quality audits. Operators can use the interface to examine deviations, compare cycles, or conduct detailed performance analyses without relying on external software tools.

Depending on the result the operator can decide either to apply the new calibartion values directly into the motion system, or to generate a formal protocol of the measurment.These actions are accessible via clearly labeled buttons in the interface. Figure 4.4 illustrating the post-measurement logic in the MMI.

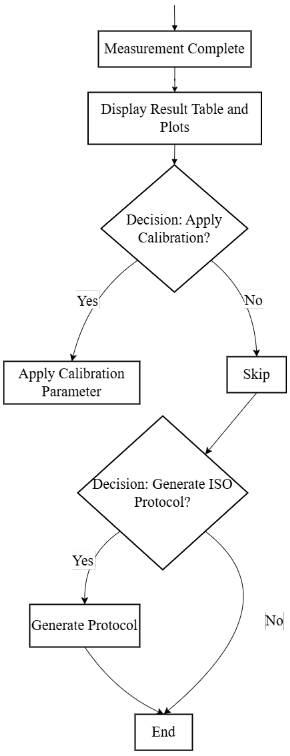

Measurement Complete
Display Result Table and Plots
Decision: Apply Calibration?
Yes
No
Apply Calibration Parameter
Skip
Decision: Generate ISO Protocol?
Yes
No
Generate Protocol
End

Figure 4.4.: Post-measurement logic for calibration and protocol handling.

Overall, the MMI has been designed with usability and flexibility in mind. Whether you're an experienced engineer or a production technician, the interface makes it easy to run tests, monitor performance, and understand results all without needing to manually process data outside the system.

4.2.5. Reporting Module

The Reporting Module represents the concluding phase of the 1D measurement workflow and is seamlessly integrated into the microSTAGE environment. This module is designed to produce a meticulously structured, traceable, and professionally formatted report that consolidates all pertinent data gathered during the measurement cycle. Such a report serves dual purposes it acts as technical documentation and as a quality assurance artifact compliant with internal company standards.

Once the Python-based analysis module finishes processing the data, it sends back the results in a structured JSON format. Then using HTML and JavaScript in microSTAGE to dynamically generate the report within the application itself. This setup gives us a lot of flexibility in how the report looks and is laid out, while also making sure every document follows a consistent visual style. The report is created as an HTML file, which is then automatically converted and saved as a PDF in a user-defined output directory.

The report follows a company-specific format, ensuring consistency across projects while accommodating the unique structure and requirements of 3D-Micromac's workflows. The report is divided into several structured sections:

- General Information: Captures essential metadata, including machine name, customer details, operator, and the exact date and time of the test.
- Measurement Device Details: Records detailed information about the interferometer, such as model, calibration status, and serial number.
- Axis Configuration: Documents specifics about the axis, including name, type (linear/rotational), direction, and motion parameters like velocity, acceleration, and jerk.
- Measurement Method Summary: Summarizes key aspects of the measurement process, such as the type of movement (linear or pendulum), number of measurement points, cycles, and start/end positions.
- Core Results and Graphs: Generates graphical plots based on requirements, illustrating forward and backward movement deviations, mean paths, and bidirectional positioning trends.
- Results Table: Provides a comprehensive tabular summary of calculated metrics, such as accuracy, repeatability, reversal error, and systematic deviations.
- Environmental Conditions: Includes contextual data such as temperature, pressure, and wavelength during the measurement process.

The system also supports conditional inclusion of additional visualizations. For example, operators can decide whether or not to include graphs showing variations between individual measurement runs or overlay plot that compare data across different cycles. These customization options are available in the system's user interface before the report is generated, giving the user control over the level of detail.

Once generated, the report is saved as PDF which can be retrieved directly from a user specified folder, ensures that report are immediately available for project documentation, customer delivery, or audit purposes. By handling the entire reporting process internally, microSTAGE removes the hassle of relying on third-party tools or manually adjusting formats. This streamlined approach not only saves time but also ensures consistency, transparency, and compliance with industry standard documentation requirements.

These components work together to create a modular and cohesive architecture for automating the 1D measurement procedure in microSTAGE. Each module functions separately, yet they communicate via well-defined interfaces and data exchanges. This flexibility ensures that the system is maintained, scalable, and adaptable to future modifications, such as adding measurement dimensions or supporting multiple assessment standards.

4.3. Data Flow and Communication

The measurement automation framework operates on a sequential and tightly synchronized data flow architecture. Users define their measurement configurations using the microSTAGE MMI, specifying details like the measurement range, number of points, and cycles. Once configured, the process starts directly from this interface. The system then moves through a series of steps, ensuring each stage is thoroughly completed and validated before proceeding to the next.

The initial step involves moving the axis to the desired position, achieved by sending API commands from microSTAGE to the ACS motion controller. The motion follows predefined profiles that take into account velocity, acceleration, jerk, and soft limits. After confirming the axis has reached its target position and motion stability, microSTAGE initiates a data acquisition request.

Data is collected through an API call to the INFAS interferometer software. Instead of continuous real-time streaming, the system takes snapshots of the most recent sensor data. This includes positional data from interferometric channels along with environmental factors such as temperature, pressure, and laser wavelength. Multiple readings are taken at each measurement point, averaged to reduce noise, and stored in a structured CSV file. Before advancing to the analysis phase, the raw dataset is logged and verified.

Once the measurement cycle concludes, microSTAGE triggers the analysis process by executing a Python script via VBScript. The CSV file containing the raw data is passed as an input parameter. The Python backend calculates positional deviation, accuracy, repeatability, and systematic error metrics according to ISO 230-2 standards. The results, including numerical outputs and links to any generated visualizations (like plots or deviation curves), are returned to microSTAGE in a structured JSON format.

During execution, microSTAGE keeps detailed logs of all operations, including API interactions, data handling activities, and script execution statuses. If issues arise, such as timeouts or errors in retrieving data from the motion controller or interferometer, the system logs the problem and alerts the user to either attempt the failed step again or terminate the process.

Finally, the results are presented within the MMI for user review. Users can examine tabulated metrics and graphical representations created during the analysis. If needed, the findings can be compiled into a formal measurement protocol using the built-in HTML/JavaScript-based reporting tool, which formats the data and automatically generates a PDF for storage or distribution.

Inter-process communication in this system relies entirely on file exchanges rather than real-time messaging systems. This approach simplifies development and debugging but may limit performance and flexibility in more demanding or scalable scenarios.

Figure 4.5 illustrates the data flow architecture of the 1D measurement automation system, highlighting the sequence of interactions between the user interface, hardware modules, analysis backend, and report generation engine.

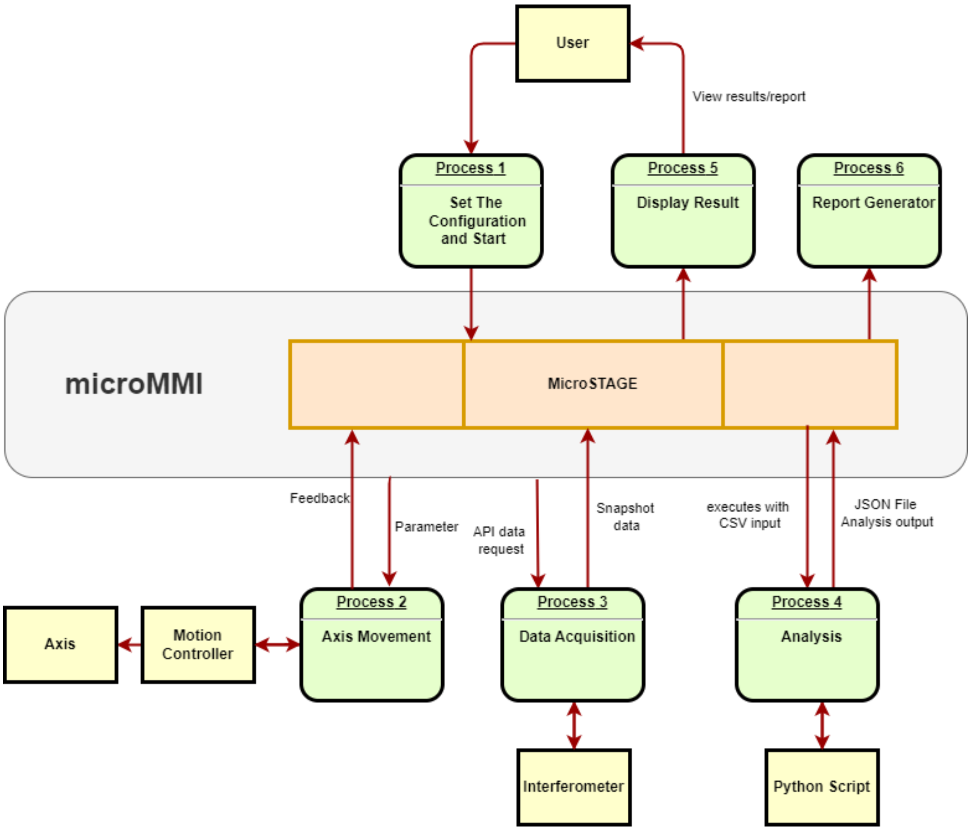


Figure 4.5.: Data Flow Architecture of the 1D Measurement Automation System

4.4. Modularity and Scalability

The automation system for 1D measurements is built with modularity at its core, ensuring ease of maintenance, reuse, and adaptability and expand over time. While the current version focuses specifically on 1D tasks using a laser interferometer and linear axis, the system architecture is built to seamlessly incorporate alternative components and measurement procedures, as long as they align with the same dimensional requirements.

The system is divided into five main software modules: motion control, data acquisition, data analysis, user interface (microSTAGE MMI), and report generation. Each of these works independently but connects effectively via clearly defined scripts or data exchanges. For instance, motion commands are executed separately from data acquisition processes but remain synchronized through coordinated scripting within the microSTAGE environment. This layered approach guarantees that modifications to one module such as replacing the interferometer or motion controller won't necessitate extensive alterations elsewhere in the system, provided compatible application programming interfaces (APIs) are present.

Internally, the modules interact through function calls and script-driven sequencing within microSTAGE. Each module processes the output of the previous one, creating a straightforward data pipeline: user configuration , axis movement , data acquisition , analysis , reporting. This layout simplifies troubleshooting and boosts traceability while maintaining operational transparency. Moreover, the VBScript routines utilized in microSTAGE are organized into reusable blocks, enabling easy adaptation or expansion of motion sequences, error management, and system initialization scripts for upcoming routines.

The Python-based analysis component, although presently confined to ISO 230-2 evaluation, is structured to accommodate future scalability. New analysis scripts adhering to diverse calibration standards or incorporating advanced statistical techniques can be developed independently and incorporated similarly through microSTAGE. This separation of analysis logic from the main execution flow empowers domain specialists to refine the backend without disturbing the frontend or automation processes.

Despite being tailored for 1D linear measurement and facing hardware limitations that hinder immediate expansion into 2D or 3D calibration, the software framework itself imposes no boundaries on expanding within its dimensional scope. The microMMI firmware underlying the microSTAGE interface accommodates a broad spectrum of motion controllers, amplifying the framework's versatility. Consequently, the same modular design can potentially be leveraged for supplementary linear measurement protocols, extended calibration cycles, or novel diagnostic functionalities.

To summarize, the system's modular architecture fosters ongoing development, streamlines upkeep, and integrates effortlessly with a variety of industrial elements. Even though it operates within specific physical measurement contexts, the software is engineered to expand both functionally and technologically within its realm.

A file-centred architecture was selected to decouple time-critical controller tasks from offline analysis while preserving full traceability. Each module exposes a narrow, well-documented interface that permits future extension to multi-axis stages or alternative sensors without refactoring the entire stack. The concept thus provides a principled groundwork for the concrete implementation reported next.

5. Implementation

Building upon the architectural blueprint, the implementation section documents the realisation of the framework inside 3D-Micromac's microSTAGE environment. VBScript wrappers, Python analysis code, and HTML/JavaScript reporting templates cooperate on a single industrial PC to drive an ACS SPiiPlus CMHP controller and a SIOS SP 2000 TR interferometer, all under version control.

5.1. System Overview

The automation framework for 1D measurements described in this thesis was developed and tested directly on a real machine setup at 3D-Micromac. The system was implemented within the company's proprietary automation environment and deployed as a custom application called microSTAGE, which runs on their broader control platform known as microMMI. This platform provides built-in tools for low-level machine control as well as the graphical interface used across all machine operations. The main objective of microSTAGE is to unify all functions outlined in the Concept chapter including motion control, data collection, analysis, calibration, and reporting into a single integrated system. This consolidation ensures that the full workflow, from initial configuration to final evaluation, is both smooth and efficient.

At its core, the system architecture consists of four main elements: a motion controller that drives the linear axis, an interferometer for acquiring high-precision positional data, a PC hosting all control and processing logic, and a set of automation scripts that coordinate the interaction between these components. All elements are deployed on a single PC, minimizing latency and simplifying deployment and testing.

For motion control, the system uses an ACS SPiiPlusCMHP controller with EtherCAT communication and three integrated motor drives. This controller interfaces directly with the microMMI environment through a native C++ connection, enabling accurate command execution and real-time feedback monitoring. Motion sequences and parameter configurations are executed within microSTAGE via VBScript, which is tightly integrated into the microMMI scripting environment.

On the measurement side, the system employs a SIOS SP 2000TR laser interferometer with three beams. It is connected to the PC via USB and interfaced using the manufacturer's API. During each measurement cycle, the device not only captures precise positional data but also logs environmental factors such as temperature, air pressure, and laser wavelength. These additional inputs support more accurate error calculation and calibration procedures.

The full hardware integration is illustrated in Figure 5.1, where the IPC communicates with both the SPiiPlusCMhp motion controller and the SIOS interferometer using Ethernet and USB, respectively.

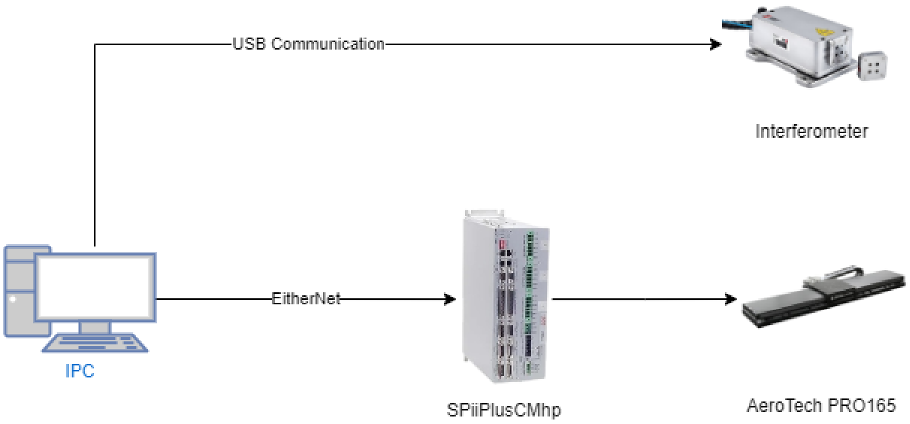


Figure 5.1.: Hardware architecture and communication layout of the measurement system. The PC (IPC) runs microSTAGE and controls the motion system via EtherCAT and reads interferometer data via USB.

All essential functionalities of motion control, data acquisition, analysis, calibration and reporting are executed on the same PC running microMMI software. The process starts when a user sets up the test parameters through the microSTAGE interface. From there, VBScript code inside microSTAGE manages the sequence of actions, initiating movements, collecting measurement data from the interferometer via its API, and transmitting the results to an external Python analysis module. This division of tasks in interface handling, motion control, and data processing makes the whole system more flexible and easier to maintain.

Data analysis is conducted using Python, selected due to its powerful scientific computing libraries and adaptability in processing structured data. The analysis logic, written in Python 3, leverages libraries like Pandas for data manipulation and Matplotlib for generating visualizations such as measurement deviations, repeatability curves, and plots. Following each measurement cycle, microSTAGE exports the gathered data into structured CSV files, which are subsequently handed over to the Python analysis script via a shell command. The script processes the data and outputs a JSON file containing computed metrics, calibration parameters, and links to generated plots. This modular division between control logic and analysis ensures that each component can progress independently without interfering with the overall workflow.

Final results are generated dynamically in HTML and JavasScript in a structure that analysis findings and user metadata into a format compliant with the company's specific protocols. then presented in microSTAGE and converted into a structured PDF and saved for documentation or distribution purposes. Throughout the development process, Git was used for version control. This ensured reliable collaboration, tracked changes systematically, and enabled isolated development and testing of different parts of the system.

5.2. Motion Control Integration

The motion control system for the 1D-Measurement framework was developed using VBScript within the microSTAGE application, taking advantage of the built-in functionalities offered by the microMMI platform. Physical control of the axis is managed by an ACS controller, which connects to the host PC via EtherCAT. This controller provides a high-level command set for managing motor drives and reading real-time feedback data.

The ACS controller support various motion types such as absolute positioning and relative moves. for this implementation absolute positioning was used for the movement and a custom homing buffer that is predefined within the ACS environment was invoked to reference the axis before any motion begin. Commands for axis control are issued through a scripting interface that abstracts low-level communication with the ACS controller. The frequent command used in the system are ”Enable” to activate the axis, ”Home” to run a custom homing buffer stored on the controller, which establishes a reliable mechanical reference before the start of any measurement cycle, and ”Move” that send the axis to a specific position. After each move, ”WaitUntilInPos” makes sure the axis has fully settled before anything else continues. These commands are all built into the scripting interface and directly control the motion controller.

Before starting any measurements, the axis needs to be initialized, This involves enabling the motor drive, check whether the axis is not homed to execute the homing routine defined in the controller, and to set the axis parameter such as velocity, acceleration and jerk. These procedures are contained in a reusable function to guarantee a secure and uniform setup for every measurement run.

```
Sub PrepareAxisToMove()
    SetAxisParameters
    Enable g_Axis
    If Not g_Axis.Homed Then
        Home g_Axis
        WaitUntilInPos g_Axis
    End If
End Sub
```

Once the axis is properly homed, the system reads the motion limits using the “GetSoftwareLimits“ command. These limits help make sure the movement of the axis is within safe boundaries. If a target position goes beyond those limits, the execution halt right away and display a warning message through the MMI. This helps avoid any risky movements that could harm the equipment.

Motion parameter like speed, acceleration, and jerk are configurable through the microSTAGE interface. When those values are set, the system applies them automatically to the axis before starting any movement. This flexibility enable the users to adjust how the axis behaves depending on how sensitive the parts are, or what kind of data needed.

The motion control logic supports two primary measurement modes: linear and pendulum movement. These modes determine how the axis traverses the measurement points and how directionality is handled across cycles.

In linear mode , the axis moves in one direction from the start to the end position, pausing at predefined points to collect data. After completing all measurement points in one direction, the axis start measure in the backward direction until it returns to the start position and begins the next cycle. This forward and backward traversal is repeated for the number of cycles defined by the user. The logic structure is as follows:

```
Sub StartLinearCycle(arrTargetPositions)
    Dim limits
    Set limits = GetAxisLimits()

    Dim nRun ,nIndex
    For nRun = 1 To nNumCycle

        MoveAxisToPosition(limits("Low") + 0.1)

        for nIndex = LBound(arrTargetPositions) To UBound(arrTargetPositions)
            MoveAxisToPosition(arrTargetPositions(nIndex))
            TriggerInterferometer
        next

        MoveAxisToPosition(limits("High") - 0.1)

        For nIndex = UBound(arrTargetPositions) To LBound(arrTargetPositions) Step -1
            MoveAxisToPosition(arrTargetPositions(nIndex))
            TriggerInterferometer
        Next
    Next
End Sub
```

In pendulum mode, the axis follows a step by step movement. Instead of moving through the entire stage in one direction and returning, it designed to measure each position in both directions before moving to the next point similar to how a pendulum swings back and forth. The axis first moves forward to the first measurement point, then backward to capture data in both directions. It then moves to the second point, and so on. This forward and backward sequence is repeated for each measurement point across the axis range. The logic structure is as follows:

```
Sub StartPendulumCycle(arrTargetPositions)
    Dim limits
    Set limits = GetAxisLimits()

    Dim nRun ,nIndex
    For nIndex = LBound(arrTargetPositions) To UBound(
   arrTargetPositions)

        For nRun = 1 To nNumCycle

            MoveAxisToPosition(arrTargetPositions(nIndex))
            TriggerInterferometer

            If nIndex = LBound(arrTargetPositions) Then
                MoveAxisToPosition(limits("Low") + 0.1)
            Else
                MoveAxisToPosition(arrTargetPositions(nIndex - 1))
                TriggerInterferometer
            End If
        Next

        If nIndex = UBound(arrTargetPositions) Then
            For nRun = 1 To nNumCycle
                MoveAxisToPosition(limits("High") - 0.1)
                wait 200
                MoveAxisToPosition(arrTargetPositions(nIndex))
                TriggerInterferometer
            Next
        End If
    Next
End Sub
```

At each measurement point, the axis is brought to position using the following motion routine. This subroutine ensures that every move is completed before data is acquired:

```
Private Sub MoveAxisToPosition(ftargetPos)
    Move g_Axis, ftargetPos
    WaitUntilInPos g_Axis
End Sub
```

The measurement cycle follows a set sequence. Once the interferometer is zeroed at the starting point, the axis begins its programmed movement. Depending on whether the user chose linear or pendulum mode, the system will move through a series of positions. At each stop, the system waits until the axis is completely still confirmed with the "WaitUntilInPos" command then captures the interferometer reading. It keeps doing this until all points have been measured, and then returns to the original starting position to finish the cycle.

By combining well structured movement logic, parameter configurability, and real-time synchronization, the system achieves high precision and repeatable motion. Its modular implementation allows future expansion with minimal disruption, such as the addition of new movement patterns or advanced error compensation strategies.

5.3. Data Acquisition Integration

Data acquisition in the 1D-Measurement automation is handled using the SIOS SP 2000 TR interferometer device, which is connected to the control PC via USB. This high precision device provides continuous, real-time feedback on positional changes using three separate laser channels, as well as environmental readings such as temperature, pressure, and laser wavelength. The interferometer is managed through its manufacturer provided software, which runs continuously in the background and exposes a COM based API interface for external access.

The measurement process is closely synchronized with axis motion. After the axis is moved to a target position, and stabilization is confirmed using the "WaitUntilInPos" command, the system begins data collection. At this point, live interferometric data is streamed through the API into the microMMI environment, where it is continuously monitored. However the script within microSTAGE samples the data only a trigger sent. This ensures that measurements reflect the actual position under stable conditions, eliminating transient effects caused by motion or vibration.

The system gathers 10 interferometric data samples over a brief period of time for every measurement point. These values are averaged directly in VBScript within microSTAGE to reduce any measurement noise, providing a clean and stable reading that reflects the true displacement. This averaged value is then passed along to the analysis module for further processing.

The main referance for tracking linear displacement is taken from channel 2, which is aligned with the motion axis. In addition, the value from all three channels are used to estimate angular deviations such as pitch and yaw by comparing the differences in measured path length. These computed values are used during analysis to identify displacement errors and create calibration paramter if needed. Although environmental variables such as temperature, pressure, and laser wavelength are also read from the interferometer, they are not directly used in the analysis calculations.

Instead, these values are monitored and displayed to the user within the MMI for informational and traceability purposes.

This structured acquisition process ensures consistency in measurement results while maintaining tight integration with the motion control logic. Furthermore, it offers flexibility for future enhancements, such as increasing the number of samples per measurement or dynamically adjusting the timing based on real-time feedback from the system.

5.4. Data Analysis

The analysis process begins immediately after completion of the measurement cycle within microSTAGE. At this stage, all interferometer and configuration data including time-stamped position readings, environmental values, and the full set of parameters for the measured points are saved to a CSV file with a fixed schema. This CSV acts as the standard input for an external Python module that performs the evaluation.

To initiate the analysis, microSTAGE invokes the Python script via a VBScript shell command, passing the CSV path as an argument. While the script is executing, the system suspends further actions and monitors the run for errors; any anomalies are logged for later review to ensure traceability.

The analysis module, implemented in Python 3 using Pandas for tabular handling, NumPy for numerical operations, and Matplotlib for visualisation, first loads the input into a DataFrame in which each row is a measurement sample and columns capture axis positions, motion direction, cycle/run indices, timestamps, and environmental conditions.

The analyser reads a time-series file containing time, three displacement channels, temperature, air pressure, and laser wavelength. Displacement units are discovered from the accompanying header and converted to micrometres; millimetre inputs are scaled by 1000, while micrometre inputs are retained. The target position at each sample is the rounded commanded location, with sign derived from the negative-direction setting. Channel 2 serves as the primary observable for ISO calculations.

ISO 230-2 requires separating approach directions. In linear cycles, direction alternates blockwise across successive target lists. In pendulum cycles, direction is inferred from the sign of the incremental displacement while excluding leading and trailing transients. Each observation is therefore labelled Forward or Backward and assigned a run index so repeated approaches to the same target can be grouped for variance estimation.

Temperature and air pressure are recorded alongside displacement. For each run, the mean, minimum, maximum, and deviation from the configured temperature reference are computed and stored to contextualise differences between runs. The interferometer remains responsible for refractive-index compensation and wavelength stability; no post-correction is applied at this stage.

5.4.1. Metric Calculations

The main goal of the analysis phase is to calculate the accuracy and repeatability of the axis, taking backlash into account. which is essential for assessing the performance of the axis. in this project, the python analysis script implements all the necessary parameters, including systematic positional deviation, bi-directional positioning accuracy, repeatability, and reversal error. Which all the calculations follow the mathemtical formulas provided in the DIN ISO 230-2.

For each measurement point, the axis performs one forward and one backward movement per cycle [46]. Depending on the number of configured cycles, multiple forward and backward values are collected. These values are grouped directionally and used to compute the following calculation:

Deviation of Position:

$$x_{ij} = P_{ij} - P_i \tag{5.1}$$

Where P_i is the reference position and P_{ij} is the actual position taken from the interofermeter

Mean unidirectional positioning deviation at a position in the forward and the backward direction:

$$\bar{x}_i \uparrow = \tfrac{1}{n} \textstyle\sum_{j=1}^{n} x_{ij} \uparrow \tag{5.2}$$

and

$$\bar{x}_i \downarrow = \tfrac{1}{n} \textstyle\sum_{j=1}^{n} x_{ij} \downarrow \tag{5.3}$$

Where the positon directions for the measurement are inicated by ↑ - forward direaction, ↓ - backward direaction.

Mean bi-directional positional deviation at a position:

$$\bar{x}_i = \frac{\bar{x}_i \uparrow + \bar{x}_i \downarrow}{2} \tag{5.4}$$

The reversal error at a given position, which reflects the positional difference between approaches from opposite directions is:

$$B_i = |\bar{x}_i \uparrow - \bar{x}_i \downarrow| \tag{5.5}$$

where the overall reversal error of the axis is the maximum of all individual reversal errors:

$$B = max[|B_i|] \tag{5.6}$$

Mean reversal error of an axis is calculated as:

$$\bar{B} = \frac{1}{m} \sum_{i=1}^{m} B_i \tag{5.7}$$

The systematic positional deviation of an axis is calculated by the differeve between the maximum and the minimum of the mean unidirectinal postion deviations for one approach direction $\bar{x_i} \uparrow$ or $\bar{x_i} \downarrow$ at any position P_i along the axis

$$E \uparrow = \max \cdot [\bar{x}_i \uparrow] - \min \cdot [\bar{x}_i \uparrow] \tag{5.8}$$

$$E \downarrow = \max \cdot [\bar{x}_i \downarrow] - \min \cdot [\bar{x}_i \downarrow] \tag{5.9}$$

and the bi-directional systematic positioning error of an axis:

$$E = \max \cdot [\bar{x}_i \uparrow ; \bar{x}_i \downarrow] - \min \cdot [\bar{x}_i \uparrow ; \bar{x}_i \downarrow] \tag{5.10}$$

Now in order to get the repeatabiltiy of an axis we need to calculate the standard deviation s_i of each unidoreactional approach at a postions P_i using standard deviation formula

$$s_i \uparrow = \sqrt{\frac{1}{n-1} \sum_{j=1}^{n} (x_{ij} \uparrow - \bar{x}_i \uparrow)^2} \tag{5.11}$$

$$s_i \downarrow = \sqrt{\frac{1}{n-1} \sum_{j=1}^{n} (x_{ij} \downarrow - \bar{x}_i \downarrow)^2} \tag{5.12}$$

Using these, the unidirectional repeatability at each point is:

$$R_i \uparrow = 4 s_i \uparrow \tag{5.13}$$

$$R_i \downarrow = 4s_i \downarrow \tag{5.14}$$

Bi-directional positioning repeatability at a position is defined as:

$$R_i = \max\left[2s_i \uparrow + 2s_i \downarrow + |B_i|\,; R_i \uparrow; R_i \downarrow\right] \tag{5.15}$$

Unidirectional positioning repeatability of an axis

$$R \uparrow = max[R_i \uparrow] \tag{5.16}$$

$$R \downarrow = max[R_i \downarrow] \tag{5.17}$$

Bi-directional positioning repeatability of an axis

$$R = max[R_i] \tag{5.18}$$

The accuracy of the axis is derived from the combination of the unidirectional systematic deviations and the standard deviation for axis repeatability of unidirectional positioning using a coverage factor of 2

$$A \uparrow = \max \cdot [\bar{x}_i \uparrow + 2s_i \uparrow] - \min \cdot [\bar{x}_i \uparrow - 2s_i \uparrow] \tag{5.19}$$

$$A \downarrow = \max \cdot [\bar{x}_i \downarrow + 2s_i \downarrow] - \min \cdot [\bar{x}_i \downarrow - 2s_i \downarrow] \tag{5.20}$$

The final bidirectional accuracy of the axis is calculated as:

$$A = \max . [\bar{x}_i \uparrow + 2s_i \uparrow; \bar{x}_i \downarrow + 2s_i \downarrow] - \min . [\bar{x}_i \uparrow - 2s_i \uparrow; \bar{x}_i \downarrow - 2s_i \downarrow] \tag{5.21}$$

These metrics are computed for each defined measurement position, and the results are stored in a structured JSON file that also contains the raw samples. This design ensures traceability and allows recalculation or extension of the evaluation logic at any time. While ISO 230-2 forms the basis of evaluation in this work, the architecture is flexible enough to accommodate other standards in future projects.

5.4.2. Visualization

As part of the analysis workflow, the analysis module produces a set of standardized plots that visualize the results of each measurement cycle. These visualizations provide immediate insight into the accuracy and repeatability of the axis, that give an intuitive representation of the statistical metrics computed during the analysis. All plots are created within the Python analysis script using the “Matplotlib“ library, and they are generated automatically.

One of the key plots is the bidirectional accuracy and repeatability plot. It shows the average deviation from the intended position at each measured point, distinguishing between movement in the forward and backward directions. It include a shaded areas around the average lines represent ±2 standard deviations, giving a clear picture of the system's repeatability and error margins in line with ISO 230-2 guidelines. On this graph, key performance indicators such as maximum systematic error, total accuracy, and reversal error.

Another helpful visualization is the individual run deviation plot, where each measurement cycle data both forward and backward is displayed as a separate curve. This plot helps to assess repeatability by showing how consistently the axis reaches each point over multiple cycles. Outlier runs, noise patterns, or mechanical inconsistencies become immediately visible in this format.

Additionally, the analysis script generates separated directional plots that compare forward and backward positioning at each point. These graphs highlight reversal behavior and backlash effects, allowing users to visually distinguish directional shifts in axis behavior. Nominal position references and average deviation lines are included for context and ISO conformity.

All visualizations such as Bidirectional accuracy and reliability, individual runs, and unidirectional graph are saved as high resolution PNG files. no manual input is needed to generate these figures; the entire visualization process happens automatically during analysis, ensuring reliable results every time. By integrating these visual directly into the analysis workflow, the system transforms raw measurement data into meaningful insights. Each test run is paired with an intuitive visual summary, promoting transparency, traceability, and adherence to quality standards.

The shaded bands shown in the bidirectional and unidirectional plots are computed as $x \pm 2s$ per target and direction, where s is the unbiased standard deviation obtained in Section 5.4.1. Annotations for E, A, M, and the worst-case R are derived directly from the point-wise and axis-level statistics defined in Section 5.4.1.

5.4.3. Output Structure

At the end of each analysis run, the module writes a compact results file and a comprehensive data file, and exports a fixed set of diagnostic plots as images. This separation supports both immediate reporting and later in-depth verification without re-measurement. The compact results file, holds only the axis-level indicators and the environmental summaries consumed by the result view and the protocol generator. The comprehensive data file, preserves the full set of processed samples, grouped statistics, bidirectional aggregates, and the position wise correction table. Plots are saved as PNG images with descriptive filenames and are embedded unmodified in the user interface and the final protocol. All artefacts are written to a user-selected output directory.

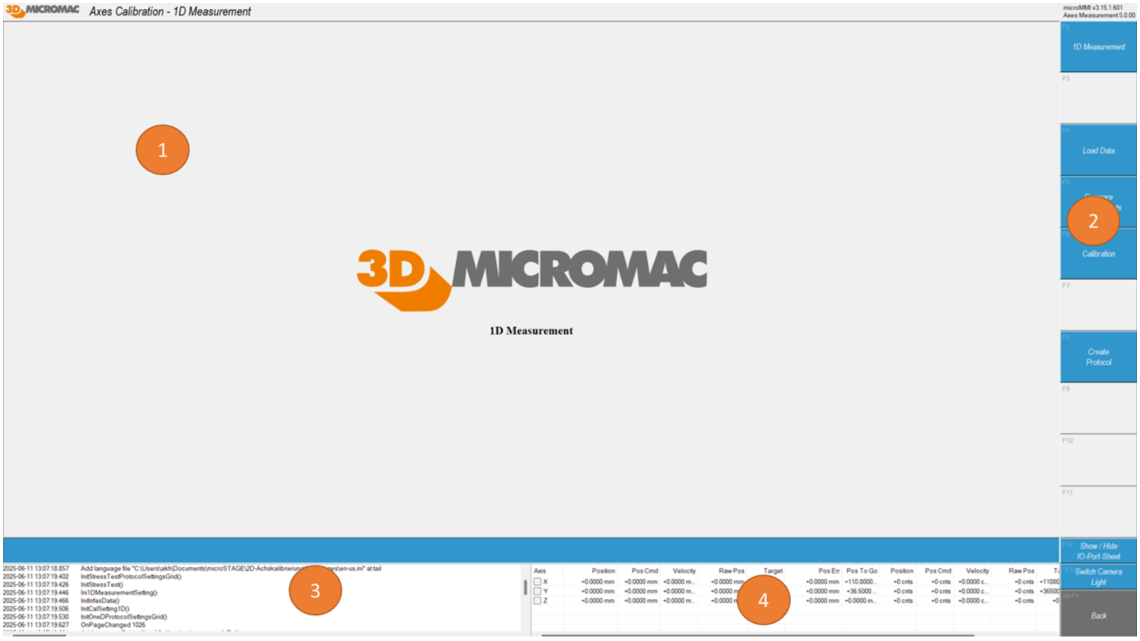
The results file structure contains two top-level objects. The first, Axis Parameters, provides the ISO 230-2 indicators as computed in Section 5.4.1 such as systematic deviation E (including direction-specific E↑,E↓), accuracy A (and A↑,A↓), repeatability R (and R↑ ,R↓), and the mean reversal value B. The second, Temperature Pressure Metrics, stores the mean, minimum, and maximum temperature and air pressure recorded over the run, together with the offset to the configured temperature reference. This compact schema is intended for protocol population and quick comparisons between runs.

The data file structure retains the full provenance of the analysis. A Processed array lists every time-stamped sample after unit normalisation, including target position, per-channel deviations, and the derived yaw/pitch estimates. A RunsData array groups the deviations by run and approach direction to support run-to-run diagnostics in the UI. Directional statistics for each target position means, standard deviations, and $x \pm 2s$ envelopes are stored under results for forward and backward subsets, the corresponding bidirectional aggregates B_i, R_i, and x_i appear under Bi_{Result}. Finally, a Correction table reports per-position compensation values in millimetres alongside the target position in millimetres, enabling downstream creation of controller correction files or calibration overlays. By persisting this structure, the framework can regenerate plots and certificates at any time without repeating the physical measurement.

The image outputs are produced during analysis using the same statistics. They include the bidirectional accuracy plot with shaded $x \pm 2s$ bands, direction-separated accuracy plots, and run-resolved deviation views. Filenames encode the plot type to allow deterministic ingestion by the result view and the protocol template. The images are stored in the same output directory as the JSON files. Using the identical data sources for on-screen results and the exported PDF ensures numerical identity between what is reviewed interactively and what is documented for audits.

5.5. User Interface (MMI Integration)

5.5.1. User Interaction and Workflow



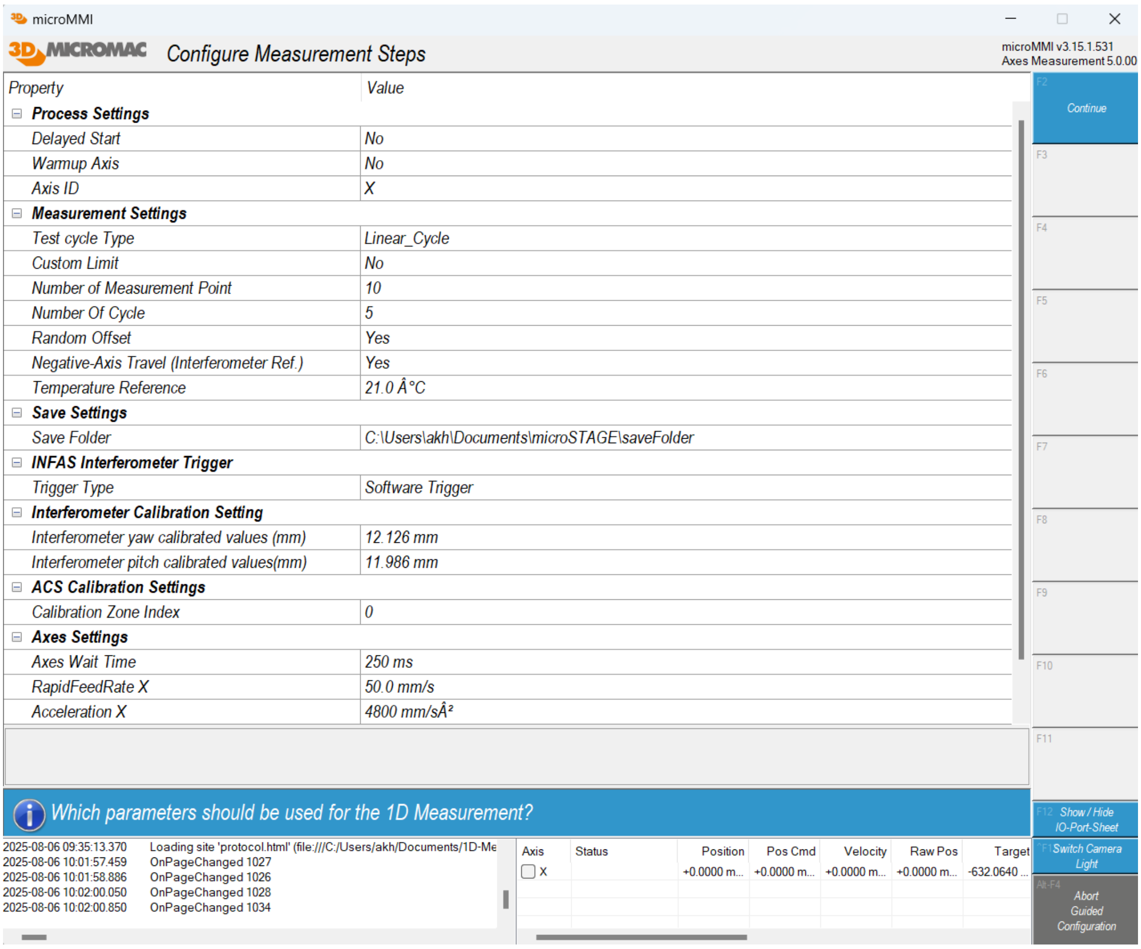
```
2025-06-11 13:07:18.857    Add language file "C:\Users\akh\Documents\microSTAGE\2D-Achskalibrierung\[illegible]ges\en-us.ini" at tail
2025-06-11 13:07:19.402    InitStressTestProtocolSettingsGrid()
2025-06-11 13:07:19.426    InitStressTest()
2025-06-11 13:07:19.446    Ini1DMeasurementSetting()
2025-06-11 13:07:19.466    InitInfasData()
2025-06-11 13:07:19.506    InitCalSetting1D()
2025-06-11 13:07:19.530    InitOneDProtocolSettingsGrid()
2025-06-11 13:07:19.627    OnPageChanged 1026
```

Axis	Position	Pos Cmd	Velocity	Raw Pos	Target	Pos Err	Pos To Go	Position	Pos Cmd	Velocity	Raw Pos	Ta
□ X	+0.0000 mm	+0.0000 mm	+0.0000 m...	+0.0000 mm	[illegible]	+0.0000 mm	+110.8000...	+0 cnts	+0 cnts	+0.0000 c...	+0 cnts	+11080
□ Y	+0.0000 mm	+0.0000 mm	+0.0000 m...	+0.0000 m	[illegible]	+0.0000 mm	+36.5000 ...	+0 cnts	+0 cnts	+0.0000 c...	+0 cnts	+36500
□ Z	+0.0000 mm	+0.0000 mm	+0.0000 m...	+0.0000	[illegible]	+0.0000 mm	+0.0000 m...	+0 cnts	+0 cnts	+0.0000 c...	+0 cnts	+0

Figure 5.2.: Overview of the MMI implementation showing configuration/display panel (1), action buttons (2), log console (3), and axis status display (4)

The control interface for the 1D measurement system is seamlessly integrated into microSTAGE, which operates on the microMMI platform. Designed with ease of use in mind, this interface allows users to configure all essential measurement parameters and initiate system operations through a straightforward, structured layout. Each key action such as starting a new measurement, loading previous results, creating a protocol, or applying calibration is executed by pressing a dedicated button as shown in area 2 in Figure 5.2. These buttons are linked to backend scripts via VBScript, ensuring precise functionality for each command.

Before any measurement starts, the user enter the configuration of the measurement in the main input section (in area 1 in Figure 5.2). the configuration includes selecting between linear or pendulum movement, setting the number of measurement points and cycles, and adjust movement settings such as velocity, acceleration, and jerk. Also specify the start and end positions for the axis. All inputs are validated against hardware limits retrieved from the controller, and any out-of-range values trigger immediate user feedback. An example shown in figure 5.3



Configure Measurement Steps

microMMI v3.15.1.531
Axes Measurement 5.0.00

Property	Value
⊟ **Process Settings**	
Delayed Start	No
Warmup Axis	No
Axis ID	X
⊟ **Measurement Settings**	
Test cycle Type	Linear_Cycle
Custom Limit	No
Number of Measurement Point	10
Number Of Cycle	5
Random Offset	Yes
Negative-Axis Travel (Interferometer Ref.)	Yes
Temperature Reference	21.0 Â°C
⊟ **Save Settings**	
Save Folder	C:\Users\akh\Documents\microSTAGE\saveFolder
⊟ **INFAS Interferometer Trigger**	
Trigger Type	Software Trigger
⊟ **Interferometer Calibration Setting**	
Interferometer yaw calibrated values (mm)	12.126 mm
Interferometer pitch calibrated values(mm)	11.986 mm
⊟ **ACS Calibration Settings**	
Calibration Zone Index	0
⊟ **Axes Settings**	
Axes Wait Time	250 ms
RapidFeedRate X	50.0 mm/s
Acceleration X	4800 mm/sÂ²

Which parameters should be used for the 1D Measurement?

```
2025-08-06 09:35:13.370    Loading site 'protocol.html' (file:///C:/Users/akh/Documents/1D-Me
2025-08-06 10:01:57.459    OnPageChanged 1027
2025-08-06 10:01:58.886    OnPageChanged 1026
2025-08-06 10:02:00.050    OnPageChanged 1028
2025-08-06 10:02:00.850    OnPageChanged 1034
```

Axis	Status	Position	Pos Cmd	Velocity	Raw Pos	Target
□ X		+0.0000 m...	+0.0000 m...	+0.0000 m...	+0.0000 m...	-632.0640 ...

Figure 5.3.: an example of the configuration panel in micrSTAGE.

Once configuration is complete, the interface activates the relevant execution controls. To maintain stability and prevent unintended modifications during the measurement process, the configuration area becomes temporarily locked upon initiation. This ensures that the measurement runs under consistent, verified conditions. During this phase, backend scripts manage the entire motion sequence, data acquisition, and transfer to the Python analysis module, as outlined earlier in this chapter.

Real-time feedback is provided throughout the measurement cycle. A dedicated log window (in area 3 in Figure 5.2) shows all runtime messages such as start, progress updates, errors, and completion events, using structured log statements from the underlying VBScript logic. In parallel, a real-time axis monitor (in area 4 in Figure 5.2) displays hardware level status, including commanded and actual position, velocity, and raw encoder data for the axis. This dual-display approach allows the user to monitor both high level process flow and low-level motion behavior in a single interface.

Overall, the design of this interface promotes safe and reliable operation. It keeps the setup and execution stages clearly separated, logs every user input for traceability, and links each function to a clearly labeled button or action. This structure reduces the chance of mistakes while still giving the user full visibility into how the system is working.

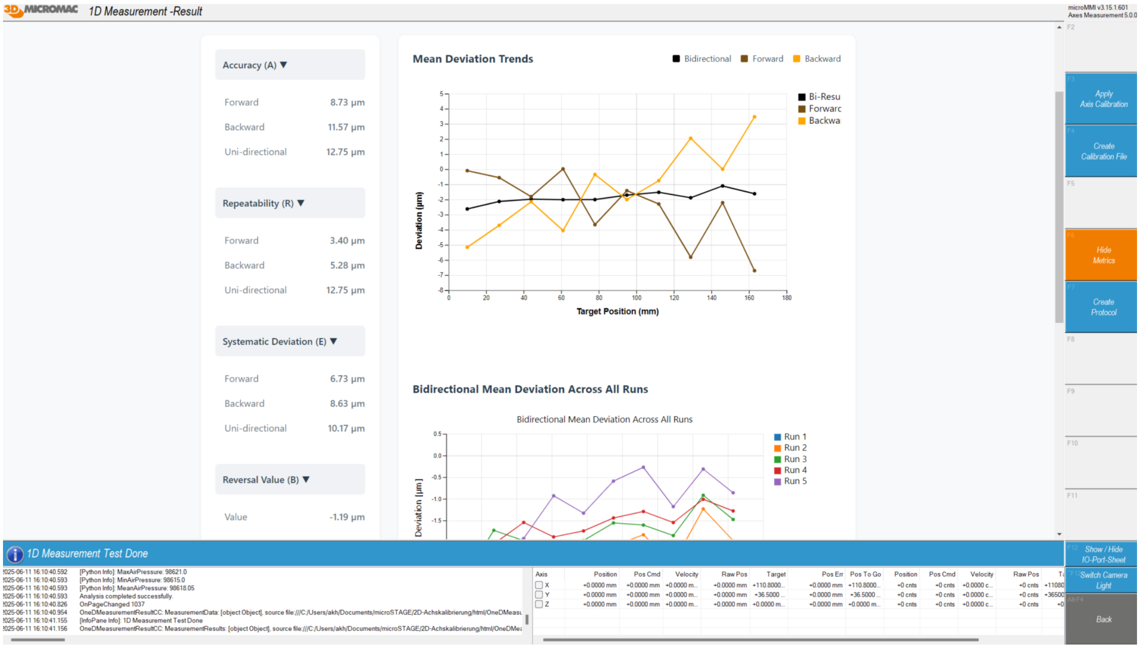
5.5.2. Result Display and Visualization

Once the analysis phase is complete and the structured JSON result is generated, the system present the final result to the user directly within the microSTAGE interface. This stage is implemented using HTML and JavaScript using D3.js Library. This result view presents both the ISO 230-2 evaluation metrics and the supporting visualizations directly in the user interface, allowing the operator to assess system performance without leaving the platform.

The visual elements in the microSTAGE result view are generated from the same statistics computed during analysis, ensuring numerical identity between on-screen values and those stored in the protocol. In particular, the uncertainty bands ($x \pm 2s$) and metric annotations (E,A,R) are read directly from the analysis output files, avoiding any duplication of logic in the UI layer.

The result view is divided into two main sections. On one side of the interface, have a structured metric table displays all relevant measurements outputs. These include the positioning accuracy A, repeatability R, systematic deviation E, and reversal error B, each broken down into forward, backward, and and bi-directional measurements. Alongside these, the table also presents environmental summaries recorded during the measurement process, such as temperature range, air pressure, and laser wavelength stability.

On the other side, several charts visually represent how the axis behaves across its full range. These include mean deviation curves, run by run comparisons, and Pitch and Yaw trend. All these visuals are powered by D3.js, which allow the user to interact with the plots like choosing which test runs to display or hiding certain ones to spot trends or irregularities more clearly. This level of interactivity enhances the interpretability of the results and supports more granular diagnosis of system behavior. Figure 5.4 shows an example of the result displayed in the MMI.



Accuracy (A) ▼

Forward	8.73 µm
Backward	11.57 µm
Uni-directional	12.75 µm

Repeatability (R) ▼

Forward	3.40 µm
Backward	5.28 µm
Uni-directional	12.75 µm

Systematic Deviation (E) ▼

Forward	6.73 µm
Backward	8.63 µm
Uni-directional	10.17 µm

Reversal Value (B) ▼

Value	-1.19 µm

1D Measurement Test Done

```
2025-06-11 16:10:40.592   [Python Info]: MaxAirPressure: 98621.0
2025-06-11 16:10:40.593   [Python Info]: MinAirPressure: 98615.0
2025-06-11 16:10:40.593   [Python Info]: MeanAirPressure: 98618.05
2025-06-11 16:10:40.593   Analysis completed successfully.
2025-06-11 16:10:40.826   OnPageChanged 1037
2025-06-11 16:10:40.954   OneDMeasurementResultCC: MeasurementData: [object Object], source file:///C:/Users/akh/Documents/microSTAGE/2D-Achskalibrierung/html/OneDMeasu
2025-06-11 16:10:41.155   [InfoPane Info]: 1D Measurement Test Done
2025-06-11 16:10:41.156   OneDMeasurementResultCC: MeasurementResults: [object Object], source file:///C:/Users/akh/Documents/microSTAGE/2D-Achskalibrierung/html/OneDMea
```

Axis	Position	Pos Cmd	Velocity	Raw Pos	Target	Pos Err	Pos To Go	Position	Pos Cmd	Velocity	Raw Pos	Ta
☐ X	+0.0000 mm	+0.0000 mm	+0.0000 m...	+0.0000 mm	+110.8000...	+0.0000 mm	+110.8000...	+0 cnts	+0 cnts	+0.0000 c...	+0 cnts	+11080
☐ Y	+0.0000 mm	+0.0000 mm	+0.0000 m...	+0.0000 mm	+36.5000 ...	+0.0000 mm	+36.5000 ...	+0 cnts	+0 cnts	+0.0000 c...	+0 cnts	+36500
☐ Z	+0.0000 mm	+0.0000 mm	+0.0000 m...	+0.0000 mm	+0.0000 m...	+0.0000 mm	+0.0000 m...	+0 cnts	+0 cnts	+0.0000 c...	+0 cnts	+0

Figure 5.4.: Result display interface in microSTAGE showing metric table and visual graphs

All of these graphs are generated automatically from the analysis data but are displayed live within the interface, eliminating the need to export or open files externally. This integrated result presentation supports quick verification of calculation result and enables direct follow up actions such as calibration or protocol generation. While the real-time viewer provides detailed insights with interactive filtering features, its layout is optimized for on-screen inspection and differs from the static format used in the final measurement protocol report.

By combining advanced automation, robust analysis, and intuitive visualization, the system empowers users with actionable insights, ensuring they not only receive data but also gain meaningful, standardized interpretations aligned with industry best practices.

5.6. Measurement Protocol

Once a measurement cycle and its analysis are complete, users can generate a formal measurement protocol directly from within microSTAGE. This report acts as an official document for verifying machine performance and is formatted to match internal company documentation requirements. The generation process is fully integrated into the automation workflow and ensures that all reported values align precisely with the latest analysis output.

The protocol is created using an internal HTML and JavaScript-based rendering engine embedded within microSTAGE. It uses a fixed template designed by 3D-Micromac, which enforces layout consistency and ensures that all key information is presented in a standardized format. The report is populated dynamically using the structured JSON file produced during analysis. Once the user start protocol generation either immediately after a measurement cycle or later by reloading a saved result the system reads the JSON content and fills the appropriate fields without requiring any manual input.

The content of the report includes:

- Measurement metadata such as machine type, customer, operator, and date
- Axis configuration and execution method, including motion type, start/end positions, and number of points
- Environmental readings collected during the measurement (e.g., temperature, pressure)
- Computed standard metrics, including bi-directional accuracy A, repeatability R, systematic deviation E, and reversal error B
- Result graphs, such as deviation trends and bidirectional comparisons, automatically embedded as static images

The report is exported as a PDF file and saved to a user-specified directory for later access, archival, or distribution. Users have the option to include or exclude specific graphs based on the intended use of the report. This flexibility allows for both simplified summaries and detailed diagnostic reports depending on the context.

Because the entire protocol is generated from the same JSON structure used for result display, it guarantees consistency between what is seen in the user interface and what is documented. Moreover, this separation of data and presentation logic enables reusability. Users can regenerate the report at any time without repeating the measurement cycle.

An example of a generated protocol report is included in Appendix B and Appendix C. It demonstrates the standard structure used by the company and reflects the values calculated during the analysis phase of this thesis.

The implementation bridges low-level EtherCAT motion control with high-level scientific-computing libraries, achieving a deterministic, fully scripted calibration pipeline. Key deliverables include reusable motion routines supporting linear and pendulum modes, a Python module that outputs ISO 230-2 metrics and plots in JSON/PNG format, and an auto-generated PDF certificate. This concrete instantiation validates the feasibility of the concept and prepares the ground for quantitative assessment.

6. Results and Evaluation

The framework is benchmarked on a 200 mm AeroTech PRO165 axis, comparing pre- and post-calibration performance as well as legacy versus automated workflows. The experimental rig, motion programme, and environmental controls are detailed to ensure reproducibility, followed by a presentation of ISO 230-2 metrics and cycle-time measurements.

6.1. Experimental Setup

A rigorous assessment of the automation framework was carried out on a dedicated calibration station configured to replicate the mechanical and environmental conditions encountered in regular production. This setup incorporates a precision linear stage, a laser interferometric measurement system with environmental probes, an industrial motion controller and a host PC running microSTAGE. The following subsections describe these components individually so that the reported performance data can be reproduced and critically evaluated.

6.1.1. Hardware Configuration

To validate the functionality and performance of the developed automation framework, the system was tested on a single linear axis used for experimental validation prior to machine integration. The test axis is equipped with an AeroTech Pro165 direct drive stage supplies 200 mm vertical travel. An iron less brushless motor drives a carriage supported by crossed-roller bearings. with calibrated geometric error is limited to $\pm 1.5\mu m$, and with encoder gives a minimum incremental motion of 5 nm.

Axis control is provided by an ACS SPiiPlus CMHP controller that closes current, velocity and position loops at 20 KHz and interpolates sin-cos feedback. Commissioning tests established a closed-loop bandwidth above 70 Hz and maintained following error below $0.2\mu m$ at the nominal test velocity of $2\,\mathrm{mm\,s^{-1}}$.
Absolute displacement is measured using a SIOS SP 2000 TR with triple beam interferometer ($\lambda = 632.8nm$, $20pm$ resolution, long term frequency stability 2×10^{-8}). Only the central distance channel is used for the measurement.
All motion control, data acquisition, and real-time processing run on an industrial PC executing microSTAGE. Interferometer phase data are sampled at 10 kHz and time stamped in software when the motion command is issued. The measured

end-to-end latency of the control chain is below 2 ms, comfortably within the dwell periods required for static captures.

A Picture of the test environment is shown in Figure 6.1 It captures the vertical AeroTech Pro165 axis used for evaluation, along with the SIOS interferometer head and the host PC running microSTAGE.

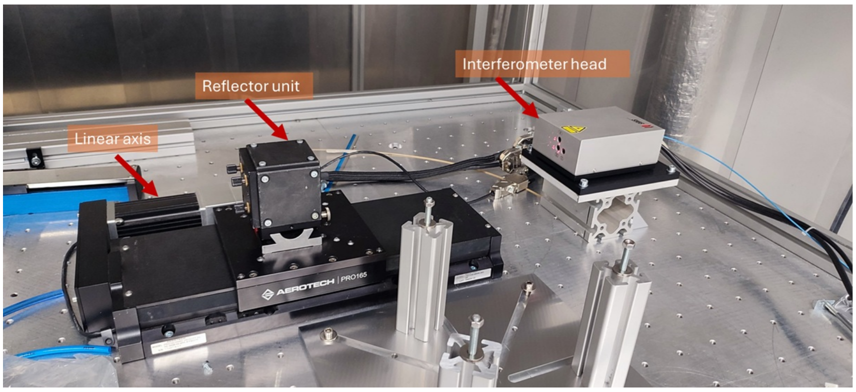


Figure 6.1.: Experimental rig comprising an AeroTech PRO165 vertical axis, a SIOS SP 2000 TR interferometer head, and the host PC running microSTAGE.

6.1.2. Environmental Control and Monitoring

The validation runs were carried out in a standard workshop environment without active temperature stabilisation or air-conditioning. Ambient variables were captured by the built-in environmental-sensor block of the SIOS SP 2000 TR interferometer, which records temperature, barometric pressure and relative humidity and streams them together with the displacement data. The sensor head was mounted approximately 50 mm from the interferometer beam path, providing representative air-path readings for wavelength compensation.

During the longest measurement sequence the temperature remained between 21.15 °C and 21.16 °C, and the air pressure stayed close to 986 hPa. Temperature, pressure and humidity samples are time-stamped by the microSTAGE and stored with every displacement record. Then it displays these values in its result viewer panel and embeds them automatically in the protocol report exported after each run.

6.1.3. Motion Programme

To characterise positioning accuracy, repeatability and reversal error, the automation framework drove the axis stage in pendulum mode, a trajectory that measures each position in both travel directions before advancing to the next an example shown in figure 6.2. Ten target points were defined randomly at 10, 27, 44, 61, 78, 95, 112, 129, 146 and 163 mm along the 200 mm stroke. At every point the carriage first approached from the lower stroke side, paused while the in-position window was satisfied, triggered the interferometer, then reversed to the preceding position or to the soft limit when the first point was reached and triggered again. This forward–backward pair is repeated five times at every position so that each recorded value captures both travel directions and any slow thermal or servo drift influences both directions equally, rather than skewing the result.

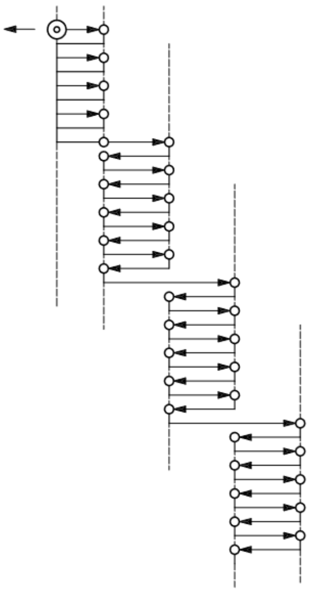


Figure 6.2.: Pendulum motion pattern, the axis visits target positions in a forward–backward pair,repeated for number of cycles

Axis moves were executed with an S-curve profile at a nominal velocity of 50 mm/s, limited to an acceleration of 4800 mm/s^2 and a jerk of 6000 mm/s^3. After motion completion the controller enforced a 6000 ms dwell, during which the interferometer collected a 6s snapshot. the samples were averaged in the microSTAGE VBScript to obtain a single displacement value for that position.

The trajectory is generated by the '*StartPendulumCycle*' routine inside microSTAGE in listing 5.2, which logs a start-of-move time-stamp for subsequent alignment with interferometer data. Velocity, acceleration and jerk parameters remain user configurable in the GUI.

6.1.4. Data-acquisition Chain

All metrological data are supplied by the SIOS SP 2000 TR interferometer, whose evaluation unit streams displacement and environmental channels over USB to the host PC. The interferometer software provided by SIOS runs continuously in the background and exposes a COM-based API. microSTAGE accesses this API through VBScript. Once the motion controller confirms the axis is within the in-position window (*WaitUntilInPos*), the script issues a trigger command that returns the most recent interferometric and sensor samples.
For each measurement point the script acquires number of samples from all three length channels and the built-in temperature, pressure and laser-wavelength sensors. the values are collected during the 6 s dwell and averaged in VBScript to suppress high-frequency noise. The resulting displacement and the averaged environmental readings are appended to a dictionary, which is flushed to a CSV file after the final cycle. After the last point is measured, microSTAGE writes the averaged results and the accompanying environmental data to a CSV file and then launches an external Python script. The CSV path is passed as a command-line argument; while the script runs, VBScript monitors its exit status. The Python module computes all ISO 230-2 metrics, generates diagnostic plots and returns a structured JSON report plus the images. microSTAGE loads these artefacts back into its result viewer and later embeds them verbatim in the automatic protocol report.

6.2. Performance Metrics

Three main criteria are used to evaluate the automation framework: the accuracy of the calculated results, the effectiveness of the entire automation process from measurement to reporting, and the overall improvement in performance over the previous manual procedures. Additionally, as part of its functional integrity, the framework's ability to support prospective future extensions and conformity to changing standards is assessed.

To evaluate geometric accuracy, the system calculates all relevant ISO 230-2 metrics that have mentioned in chapter 5. These include the systematic positioning deviations in both forward and backward directions ($E_{Forward}$ and $E_{Backward}$), the average positioning deviation (E), the positioning accuracy ($A_{Forward}$, $A_{Backward}$, A), the mean reversal deviation (B), and repeatability metrics ($R_{Forward}$, $R{-}_{Backward}$, R). These values were obtained from two comprehensive test runs, one conducted before calibration and the other after it. Each run comprised ten measurement points with five cycles pendulum movement. The automation pipeline processed the raw displacement data, conducted an ISO

230-2 analysis, and generated a structured report file that included all metrics and corresponding plots. The outputs of this automated process were then compared with a dataset independently generated in Excel. Provided that the Excel document was accurately completed and no intermediate formulas had been altered, the results obtained from both methodologies were found to be congruent across all ISO metrics, thus affirming the accuracy of the automated calculations.

The second objective of this study was to validate the capability of the complete measurement and analysis pipeline to operate independently, requiring no manual intervention after its initial setup. After the interferometer was correctly positioned and powered on, the user initiated the testing cycle via the microSTAGE interface. After that, the software autonomously managed the entire process, which included controlling the motion of the axes, activating and averaging the interferometer data, recording timestamps, and performing the analysis after the run was completed. There was no necessity for operator involvement to finalize the cycle or generate the concluding report. This autonomous behaviour confirms the success of the automation goal from both a functional and a usability standpoint.

The third evaluation axis focuses on performance improvements, particularly the time savings achieved by replacing the manual data-handling and calculation steps. The previous workflow required roughly two hours to complete a 10-point, 5-cycle pendulum test, including Excel processing. With the new automation framework, the entire measurement cycle, including data capture, averaging, and report generation, was consistently completed within 15 to 20 minutes. The total duration varies depending on the number of positions and cycles configured, but in all observed cases the process completed well within the 30 minute target. This represents a substantial improvement in throughput and operator efficiency.

In addition to replicating the ISO metric results and reducing total cycle time, the automation also enabled more consistent calibration validation. After a compensation table was applied to the axis controller, the ISO metrics showed measurable improvement. For example, the value of A, the bidirectional deviation, decreased from 23.5 µm to 14.7 µm, which represents an improvement of approximately 37.45 percent. While the full numerical details of this change are presented in 6.3, the performance metric confirms that the automated pipeline can reliably detect and quantify calibration effects.

Lastly, the system's design places a strong emphasis on extensibility. With little need for significant reorganization, the codebase modular design makes it easier to adopt any standards or integrate new metric computations. Important features like plotting, data parsing, and unit conversion are kept apart from the logic unique to each test. Because of this structural approach, the framework is especially flexible for future improvements and applicable in a range of measurement contexts.

6.3. Data and Observations

Two complete measurement runs were carried out to evaluate the positioning accuracy and repeatability of the PRO165LM linear axis. The first run captured the system's behaviour before any calibration was applied. the second was performed after uploading the generated correction file to the controller. Each run used the same pendulum motion strategy described earlier, with ten target positions distributed along the 200 mm stroke and five forward–reverse cycles at each point. All data were processed automatically by the microSTAGE framework and the resulting ISO 230-2 metrics were extracted from the structured report output.

Figure 6.3 shows the bidirectional deviation and repeatability profile recorded during the first run. The black curve traces the average of the forward and backward positioning errors across the stroke. The shaded areas indicate repeatability, and the ISO 230-2 summary metrics are visualised using annotation arrows. Before calibration, the systematic deviation reached up to 22.15 µm (E), and the bidirectional deviation was 23.53 µm (A). Reversal error was consistently negative, with a mean value of –8.58 µm, suggesting a persistent offset between the forward and reverse paths. The repeatability margin (R) reached approximately 10.5 µm at the end of the travel.

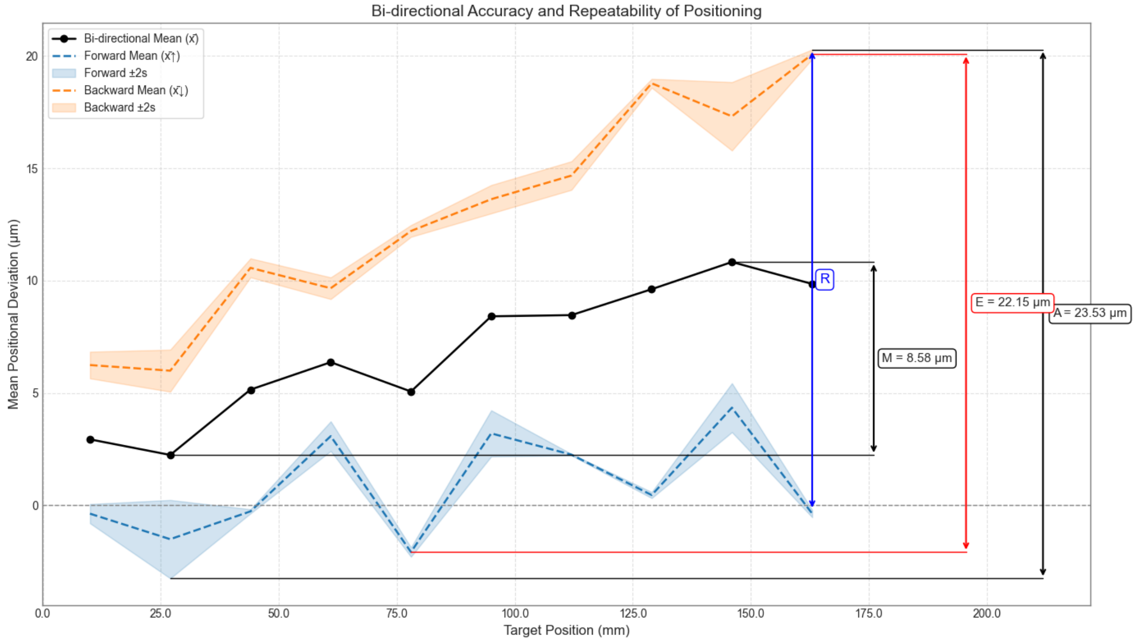


Figure 6.3.: Bidirectional positioning deviation and repeatability before calibration.

Figure 6.4 provides a full-cycle view of all five repetitions at each measurement point for both directions. While the spread of deviations is consistent from cycle to cycle, a distinct offset remains between forward and backward paths, confirming the presence of systematic reversal error in the pre-calibrated state.

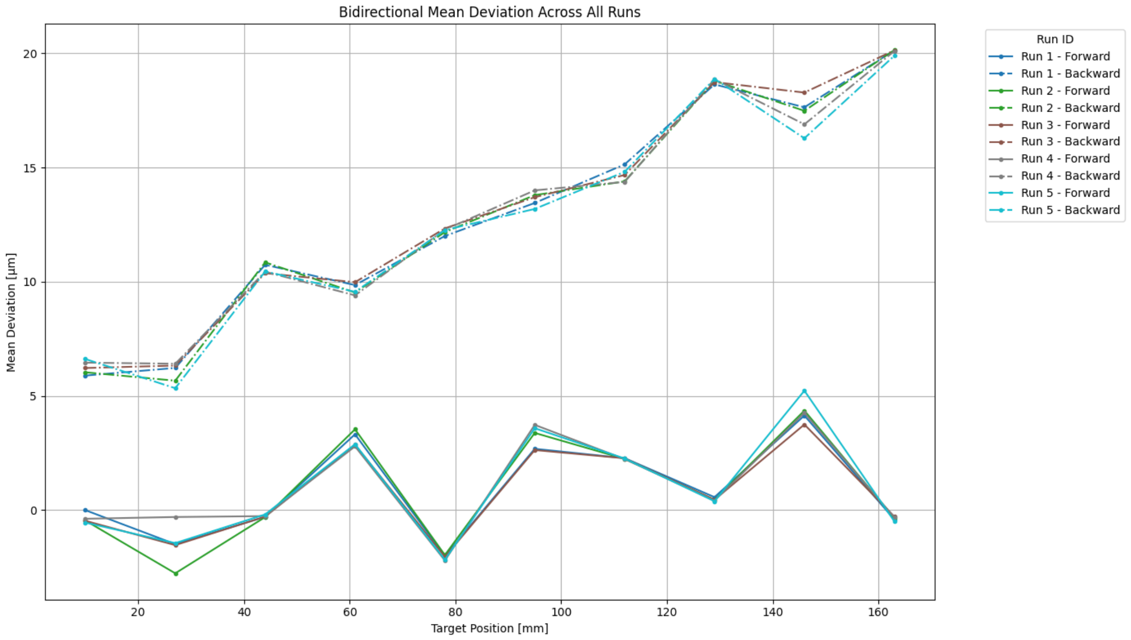


Figure 6.4.: Forward and backward deviation across all five cycles before calibration

After calibration, the profile changes considerably. As shown in Figure 6.5, the magnitude of the systematic and reversal errors decreases significantly. The average bidirectional deviation drops to 14.74 µm, and the systematic positioning error is reduced to 14.49 µm. The reversal error improves from −8.58 µm to −1.33 µm, indicating a much smaller offset between approach directions. Forward and backward deviations now align more closely across most of the stroke, and the residual repeatability band has narrowed. This is further confirmed by the cycle-level breakdown in Figure 6.6, where the dispersion of error across runs is visibly more compact and directionally symmetric.

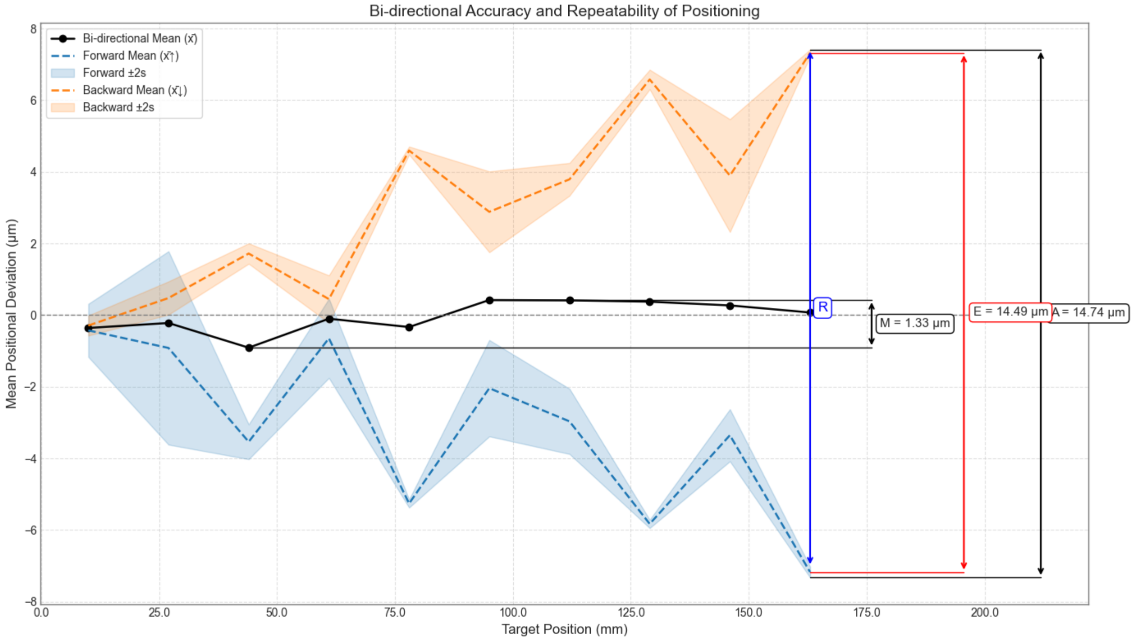


Figure 6.5.: Bidirectional positioning deviation and repeatability after calibration

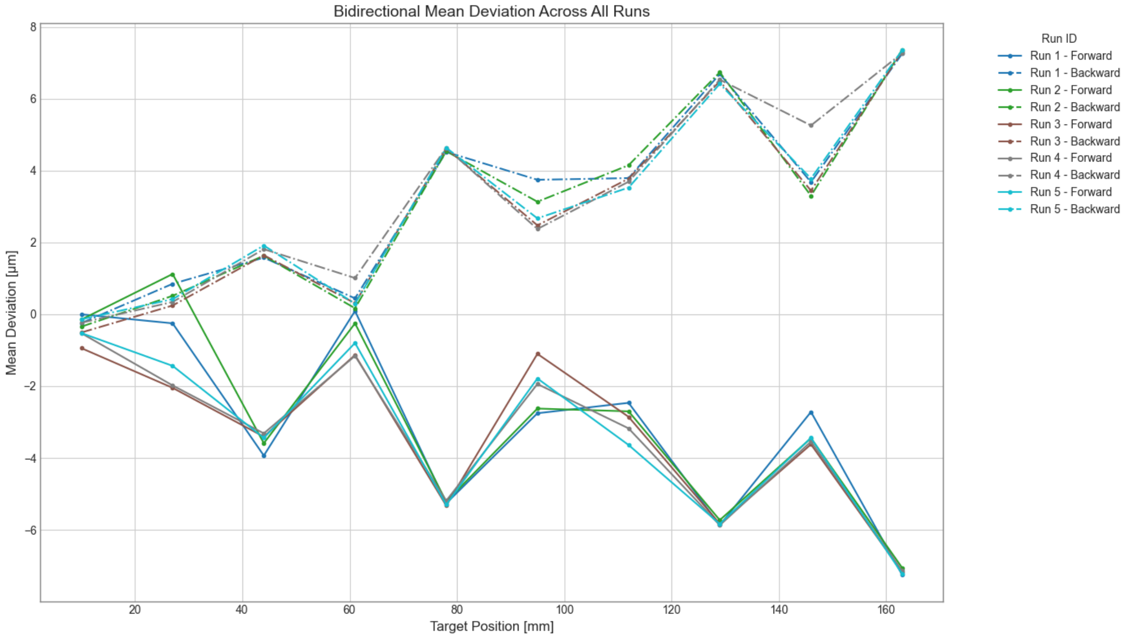


Figure 6.6.: Forward and backward deviation across all five cycles after calibration

The visual evidence provides additional insight into how errors are distributed along the travel and how calibration affects different components of the positioning behaviour. Before calibration (Figures 6.3 and 6.4), the bidirectional mean deviation exhibits a slowly varying trend with superimposed local structure near the stroke ends. This shape is consistent with a mixture of scale error and higher-order geometric effects; the outer envelope is widened where the direction-dependent means separate most strongly, indicating that reversal dominates the total uncertainty at those locations. After calibration (Figures 6.5 and 6.6), the bidirectional mean flattens and the outer envelope contracts, showing that the dominant component removed by the correction table is the systematic part of the error. Where residual widening remains, it aligns with positions that also show the largest per-position bidirectional repeatability R_i, suggesting that local dynamic effects or frictional transitions still contribute to spread even after static compensation.

A closer look at the direction-separated curves clarifies the reversal behaviour. In the pre-calibrated run, the forward and backward means maintain an almost constant offset across broad regions of travel, which appears in the bidirectional plots as a persistent negative reversal value. The post-calibration data show a marked reduction of this offset, and the forward/backward curves frequently intersect at or near zero deviation, particularly in the central travel region. This points to improved symmetry of approach and a reduction of backlash-like effects at the working speeds used for the test. The remaining asymmetry is spatially confined and coincides with transitions in the motion programme where acceleration sign change.; Under such conditions, servo loop dynamics and stiction can transiently increase R_i even if the mean error is corrected.

Run-to-run stability can be assessed by stacking all cycles. The “individual runs” panel view (each cycle plotted separately) shows that the per-run mean deviation curves are nearly parallel before calibration, confirming a reproducible systematic component. After calibration, these curves collapse more tightly onto the nominal axis, and their mutual spread narrows, indicating an overall reduction in dispersion. The analysis module also composes an aggregate overlay of the bidirectional mean across runs (“Bidirectional All Runs”) and a complementary overlay that separates forward and backward traces per run (“Backward and Forward Across All Runs”). These views corroborate that improvements are not driven by a single cycle but are consistent across repetitions, while also making any outlier run immediately visible for investigation.

Angular parasitics provide a plausible mechanism for the spatial structure observed in reversal and repeatability. The triple-beam geometry yields estimates of yaw and pitch from inter-channel differences over known baselines; these angles are logged alongside displacement for correlation analysis. Positions where reversal peaks often coincide with local extrema in the angular traces, implying that small rigid-body rotations of the carriage may alter the effective metrology line and the servo's braking/settling behaviour at direction changes. Although the present workflow does not apply angular compensation, the availability of these signals helps distinguish mechanical alignment issues from controller tuning effects and motivates targeted adjustments in future work.

Environmental stability during the runs was adequate for the purposes of comparative evaluation. Temperature and pressure are recorded and summarised (mean, minimum, maximum, and deviation from the configured temperature reference) and are included with the results to contextualise any drift between runs. Because refractive-index compensation is handled by the interferometer, these summaries primarily serve auditability and to flag atypical conditions rather than to drive additional corrections at this stage.

Finally, the effect of the correction table can be interpreted in terms of residual error structure. The systematic component, captured by the range of the directional means, reduces substantially after calibration, as reflected by the lower EE and AA values. Residuals that persist are largely local and likely tied to higher-order non-linearities in the drive train or to dynamic settle times at the chosen acceleration/jerk settings. This separation—global scale/linearity corrected, local dynamics remaining—supports the architecture's design choice to apply static compensation from measurement data and reserve any dynamic compensation or angular-error mapping for future extensions.

Table 6.1 summarises the key ISO 230-2 metrics before and after calibration. The values confirm a clear improvement in every evaluated category. The bidirectional deviation A was reduced by approximately 37.4%, and the systematic positioning error E improved by 34.5%. Reversal error and repeatability also showed significant improvements. All outputs were produced by the same microSTAGE automation script, confirming consistent post-processing and repeatability of the software framework.

Table 6.1.: Comparison of ISO 230-2 metrics before and after calibration

Metric	**Before Calibration (µm)**	**After Calibration (µm)**
E_{Forward}	6.42	6.75
E_{Backward}	14.08	7.61
E	22.15	14.49
B	-12.04	-6.36
R_{Forward}	3.49	5.40
R_{Backward}	3.04	3.15
R	20.81	14.74
A_{Forward}	8.68	9.11
A_{Backward}	15.23	7.99
Λ	23.53	14.74

The full microSTAGE protocol reports from both runs are included in Appendix B and Appendix C. These confirm that the same automated pipeline was used in both cases without manual intervention, and that the test setup remained unchanged between the two measurements.

6.4. Evaluation of Accuracy and Repeatability

The results presented in 6.3 confirm that the automated framework successfully replicates the geometric measurement process defined by ISO 230-2 and that it produces accurate, repeatable results in agreement with the previously validated Excel-based workflow. No numerical discrepancies were found between the manual and automated calculations, provided that the Excel formulas remained unmodified during data entry. This consistency verifies the correctness of the automated analysis pipeline implemented in Python and integrated with microSTAGE.

The comparison of ISO 230-2 metrics before and after calibration reveals a measurable improvement in positioning accuracy. The systematic deviation E was reduced from 22.15 µm to 14.49 µm, and the bidirectional deviation A decreased by over 37%, from 23.53 µm to 14.74 µm. This demonstrates that the calibration table applied to the controller was correctly interpreted and that the automation framework is sensitive enough to detect changes in the underlying axis behaviour. The reversal error B, which reflects the positional offset between forward and backward travel, was also reduced by nearly half, confirming that backlash or servo asymmetry was partially compensated by the calibration.

Repeatability results were more complex. While the backward repeatability improved slightly, the forward repeatability margin increased after calibration. This suggests that the calibration function introduced some small non-linearity or local overcompensation in the forward direction. However, the overall repeatability range was still lower in the post-calibration run, and the alignment of both directional curves improved substantially, as shown in Figures 6.5 and 6.6.

In addition to metrological performance, the automation framework succeeded in meeting its functional objectives. Once the interferometer was set up, the system performed the complete measurement sequence, data acquisition, analysis, and reporting without any manual intervention. All steps were triggered from within microSTAGE and executed in sequence, with internal validation at each stage. This level of integration significantly improves usability, reduces the chance of operator error, and shortens the test cycle duration.

From a practical standpoint, the cycle time improvement is among the most notable outcomes. The full process, which previously took up to two hours using manual scripting and Excel sheets, was reduced to under 20 minutes for the same number of points and cycles. This gain is especially significant in production or lab environments where multiple axes must be verified regularly. Furthermore, the modular structure of the codebase allows additional ISO metrics or even custom standards to be integrated by modifying only the analysis layer, without restructuring the full measurement logic. This confirms the system's extensibility and suitability for future use.

In summary, the developed automation framework achieves reliable accuracy, consistent processing, and significant time reduction while maintaining compatibility with industry-standard measurement principles. The results validate the core design decisions and justify the automation effort for 1D positioning verification using heterodyne interferometry.

The automated pipeline reproduces legacy spreadsheet calculations while shrinking a 10-point, five-cycle test from roughly 120 min to 15–20 min, an 85 % reduction. Bidirectional deviation (A) falls from 23.5 µm to 14.7 µm, and systematic error (E) from 22.1 µm to 14.5 µm after applying the generated correction table; repeatability and reversal error improve in tandem. These findings confirm that the framework accelerates throughput without compromising and in some respects enhancing measurement fidelity.

7. Discussion

The empirical findings are interpreted against the original objectives. Particular attention is given to residual errors after calibration, the influence of thermal drift and control-loop dynamics, and architectural trade-offs arising from the chosen file-based approach. Limitations and potential extensions including thermal compensation and multi-axis generalisation are analysed.

7.1. Summary of Key Outcomes

The experimental data in Chapter 6 show that the automation is possible to replicate a full standard compliant displacement verification process with minimal user interaction and consistent metrological results. The system executed every part of the measurement cycle axis motion, interferometric data acquisition, environmental logging, analysis, and protocol generation—without requiring manual intervention after the initial setup.

Comparisons between the new framework and the previously used Excel-based method showed that the automated outputs were identical to those generated by hand, provided that no changes occurred in the Excel sheet during use. This confirms that the new system performs accurate computations under stable conditions and can be trusted to replace manual workflows.

Significant reductions in cycle time were also achieved. While the manual procedure required up to two hours per run, the automated framework completed full measurement cycles in 15–20 minutes, including the report generation phase. These gains in efficiency are particularly relevant for production environments where throughput and repeatability are essential.

The software architecture was designed with modularity in mind. All measurement logic, analysis functions, and report-generation modules are decoupled, allowing new ISO metrics, movement patterns, or test sequences to be added without restructuring the entire system. This flexibility positions the framework for long-term reuse and adaptation in other machine configurations

7.2. Limitations and Technical Constraints

The presented framework deliberately focuses on one-dimensional positioning tests executed with a displacement interferometer and evaluated to ISO 230-2. This scope enables a deterministic, script-driven pipeline but imposes several practical constraints that should be considered when interpreting the results and planning extensions.

First, the environmental model is limited to the interferometer's internal compensation. Temperature and pressure are recorded and summarised for auditability, and the offset to a configured temperature reference is provided; however, no external refractive-index model or long-term drift correction is applied within the analysis layer. Under benign laboratory conditions this is adequate for comparative evaluation before/after calibration, but it limits traceable uncertainty statements without an explicit propagation model. The stored environmental summaries in results.json and the protocol serve to flag atypical conditions rather than to drive additional corrections.

Second, the geometric observability is constrained by the single-axis design. Channel 2 is treated as the primary displacement observable for ISO metrics; yaw and pitch are inferred from inter-channel differences over fixed baselines and logged for correlation rather than compensation. Residual angular parasitics therefore manifest as spatial structure in reversal and repeatability, especially near direction changes. The current workflow does not model Abbe offsets or apply angle-dependent corrections; multi-axis error separation and dynamic angular compensation are left for future work.

Third, the direction inference and grouping in pendulum mode assumes clean monotonic segments between the entry and exit of each target window. Although the implementation excludes the leading and trailing transients and labels forward/backward groups consistently, strong stick–slip or controller recovery events could blur this separation and inflate R_i. This risk is mitigated by dwell-based averaging and explicit in-position checks, but reproducibility still relies on a well-tuned motion programme.

Fourth, the integration boundary currently targets axes controlled via microMMI/microSTAGE APIs. The framework expects controllers that expose motion functions through this interface; immediate portability to heterogeneous control stacks requires either adoption of a similar control interface or an adapter layer that maps controller primitives (homing, absolute/relative moves, status) to the existing contracts.

Fifth, processing architecture couples responsiveness and determinism. microSTAGE launches the Python analysis as an external process and blocks further UI operations until completion, guaranteeing that the UI and the protocol consume the same immutable artefacts. While this eliminates race conditions and re-calculation drift, it also limits interactivity during long analyses and assumes sufficient local compute and disk I/O bandwidth. On the positive side, this separation ensures that the result viewer and the PDF protocol embed the exact numbers and images produced by the analysis, preserving numerical identity for audits.

Finally, the compensation model is static and position-based. The correction table is derived from the mean deviation for each position and applied as a feed-forward map in the controller. The process cannot adapt dynamically during execution to apply new corrections in response to observed error behaviour. This restricts the framework's application to scenarios where continuous compensation or adaptive control would be advantageous.

In sum, the framework attains its stated goals automation, reproducibility, and ISO-conformant reporting within a well-defined operating envelope. Extending that envelope will require environmental uncertainty modelling, multi-degree-of-freedom error separation, and selective relaxation of the blocking analysis model to enable interactive what-if evaluation without compromising traceability.

8. Conclusion and Future Work

8.1. Conclusion

This thesis set out to automate one-dimensional displacement verification in accordance with ISO 230-2, replacing a labour-intensive Excel workflow that demanded continuous operator attention and spanned roughly two hours per axis. The work has delivered a modular software framework that integrates motion control, interferometric data acquisition, signal processing, metric evaluation and protocol generation inside the existing microSTAGE environment. Experimental validation on a PRO165LM linear stage demonstrated that the automated pipeline reproduces all ISO metrics calculated by the former Excel sheet to within numerical identity, provided that the manual spreadsheet remains unaltered during data entry.

Beyond numerical correctness, the framework achieved a marked operational advantage: complete measurement cycles, including report generation, were consistently completed in 15–20 minutes for a ten-point, five-cycle pendulum test an efficiency gain of almost an order of magnitude. The system operated unattended after the initial interferometer setup, confirming that the chosen event-driven architecture and COM-based data exchange are robust under routine laboratory conditions.

The central contribution of the thesis therefore lies in demonstrating that ISO-compliant accuracy need not be sacrificed to achieve high measurement throughput and hands-off execution. By adopting a clean separation between motion logic, data handling and analytical functions, the implementation provides a reusable template for similar verification tasks at 3D-Micromac and, more broadly, in precision-motion laboratories that rely on heterodyne interferometry.

8.2. Future Work

Although the current framework meets its primary objectives, three avenues remain for technical refinement and broader applicability. The first concerns environmental compensation. While the interferometer corrects for air-path variations in real time, the framework presently assumes a thermally invariant axis. Incorporating material-specific expansion models would extend traceability to environments with wider temperature excursions and would align the software with emerging ISO annexes on thermal behaviour.

The second extension targets measurement scope. The modular analysis layer already supports displacement metrics, expanding it to include angular deviation and straightness would turn the tool into a general-purpose calibration suite. The motion engine can generate the required step-scan and continuous-scan trajectories, so only additional signal channels and analytical routines need to be integrated.

A third development path addresses system integration. At present the software interfaces natively only with axes controlled through microMMI. Introducing an abstraction layer would decouple the framework from a single vendor and open it to multi-axis machines, rotary tables or scanner platforms that expose different command sets. Such generalisation would also enable real-time calibration, in which updated correction tables are applied on-the-fly as error trends emerge during a run.

Pursuing these enhancements will not alter the core architecture established in this thesis. Instead, they build on the deliberate modularity that separates measurement scripts from analysis kernels and preserves clear data interfaces. Consequently, the automation framework presented here offers both an immediate productivity gain for displacement verification and a scalable foundation for future metrology applications within 3D-Micromac.

Bibliography

[1] W. M. Steen and J. Mazumder, *Laser Cutting, Drilling and Piercing.* London: Springer London, 2010, pp. 131–198. [Online]. Available: https://doi.org/10.1007/978-1-84996-062-5_4

[2] J. Bliedtner, H. Müller, and A. Barz, *Lasermaterialbearbeitung : Grundlagen, Verfahren, Anwendungen, Beispiele / Jens Bliedtner, Hartmut Müller, Andrea Barz.* München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2013, pp. 140–141.

[3] Q. Tang, S. Yin, Y. Zhang, and J. Wu, "A tool vector control for laser additive manufacturing in five-axis configuration," *The International Journal of Advanced Manufacturing Technology*, vol. 98, no. 5-8, pp. 1671–1684, 2018.

[4] E. Golubovic, "Design and realization of laser micromachining system," Ph.D. dissertation, 2011. [Online]. Available: https://research.sabanciuniv.edu/id/eprint/24564/1/EdinGolubovic_411755.pdf

[5] W. Gao, S. Kim, H. Bosse, H. Haitjema, Y. Chen, X. Lu, W. Knapp, A. Weckenmann, W. Estler, and H. Kunzmann, "Measurement technologies for precision positioning," *CIRP Annals*, vol. 64, no. 2, pp. 773–796, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0007850615001481

[6] L. Rapp, S. Madden, A. V. Rode, L. J. Walsh, H. Spallek, Q. Nguyen, V. Dau, P. Woodfield, D. Dao, O. Zuaiter *et al.*, "Anesthetic-, irrigation-and pain-free dentistry? the case for a femtosecond laser enabled intraoral robotic device," *Frontiers in Dental Medicine*, vol. 3, p. 976097, 2022.

[7] C. Brecher and M. Weck, *Automation of Machines and Plants.* Wiesbaden: Springer Fachmedien Wiesbaden, 2022, pp. 265–283. [Online]. Available: https://doi.org/10.1007/978-3-658-34622-5_7

[8] J. Schille, L. Schneider, and U. Loeschner, "Process optimization in high-average-power ultrashort pulse laser microfabrication: how laser process parameters influence efficiency, throughput and quality," *Appl. Phys. A Mater. Sci. Process.*, vol. 120, no. 3, pp. 847–855, Sep. 2015.

[9] M. L. E. Shaltout and H. Arafa, "Design of high performance feed drive systems for machine tools," Ph.D. dissertation, Faculty of Engineering at Cairo University in Partial Fulfilment of the . . . , 2009.

[10] M. A. Mohan, “A new compensation technique for backlash in position control systems with elasticity,” in *2007 Thirty-Ninth Southeastern Symposium on System Theory*, 2007, pp. 244–248.

[11] F. Paredes, C. Herrojo, and F. Martín, “Position sensors for industrial applications based on electromagnetic encoders,” *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2738

[12] N. K. Harris, J. Cronin, K.-L. Taylor, J. Boris, and J. Sheppard, “Understanding position transducer technology for strength and conditioning practitioners,” *Strength & Conditioning Journal*, vol. 32, no. 4, pp. 66–79, 2010.

[13] R. Leach, M. Ferrucci, and H. Haitjema, *Dimensional Metrology.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 1–11. [Online]. Available: https://doi.org/10.1007/978-3-642-35950-7_16871-1

[14] R. Klobucar, B. M. Strbac, M. Hadzistevic, and B. Acko, “Metrological set-up for calibrating 1d line scales with sub-micrometre precision,” *Proceedings of the 28th International DAAAM Symposium 2017*, vol. 1, pp. 0537–0544, 2017. [Online]. Available: https://www.academia.edu/68839635/Metrological_set_up_for_Calibrating_1D_Line_Scales_with_Sub_Micrometre_Precision#loswp-work-container

[15] A. Plant and R. Hanisch, “Reproducibility in Science: A Metrology Perspective,” *Harvard Data Science Review*, vol. 2, no. 4, dec 16 2020, https://hdsr.mitpress.mit.edu/pub/0r4v4k4z.

[16] P. Dowd, “Accuracy and precision,” in *Encyclopedia of Mathematical Geosciences.* Springer, 2022, pp. 1–4.

[17] A. Archenti and T. Laspas, *Accuracy and Performance Analysis of Machine Tools.* Singapore: Springer Singapore, 2019, pp. 1–30. [Online]. Available: https://doi.org/10.1007/978-981-10-4912-5_7-1

[18] P. Sigron, I. Aschwanden, and M. Bambach, “Compensation of geometric, backlash, and thermal drift errors using a universal industrial robot model,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 6615–6627, 2024.

[19] A. Le Reun, K. Subrin, A. Dubois, and S. Garnier, “Thermal drift and backlash issues for industrial robots positioning performance,” *Robotica*, vol. 40, no. 9, pp. 2933–2952, Sep. 2022.

[20] B. Karba and N. Yıldırım, “Backlash prediction of a spur gear pair with manufacturing errors of runouts, span and pitch deviations,” in *Proceedings of the International Congress on Mathematics, Engineering, Natural and Medical Sciences, EJONS V*, 2018, pp. 50–66.

[21] M. Noori and W. A. Altabey, “Hysteresis in engineering systems,” *Applied Sciences*, vol. 12, no. 19, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/19/9428

[22] Y. Cai, L. Wang, Y. Liu, C. Li, and K.-C. Fan, “Accuracy improvement of linear stages using on-machine geometric error measurement system and error transformation model,” *Opt. Express*, vol. 30, no. 5, pp. 7539–7550, Feb 2022. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-30-5-7539

[23] Y. Altintas, C. Brecher, L. Uriarte, and G. Pritschow, “Machine tool feed drives,” *Cirp Annals-manufacturing Technology - CIRP ANN-MANUF TECHNOL*, vol. 60, pp. 779–796, 12 2011.

[24] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, “A review of pid control, tuning methods and applications,” *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818–827, Jun 2021. [Online]. Available: https://doi.org/10.1007/s40435-020-00665-4

[25] M. Rashid, I. Hajizadeh, S. Samadi, M. Sevil, N. Hobbs, R. Brandt, and A. Cinar, “Chapter 15 - automated closed-loop insulin delivery: system components, performance, and limitations,” in *Glucose Monitoring Devices*, C. Fabris and B. Kovatchev, Eds. Academic Press, 2020, pp. 293–326. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128167144000156

[26] P. Welander, “Understanding derivative in pid control,” *Control Engineering -Highlands Ranch- Cahners then Reed Business Information-*, vol. 57, pp. 24–27, 02 2010.

[27] M. J. Blondin, *Optimization Algorithms in Control Systems*. Cham: Springer International Publishing, 2021, pp. 1–9. [Online]. Available: https://doi.org/10.1007/978-3-030-64541-0_1

[28] S. Palani, *Frequency Response Analysis*. Cham: Springer International Publishing, 2022, pp. 327–500. [Online]. Available: https://doi.org/10.1007/978-3-030-93445-3_3

[29] Y. Lee, S. Park, and M. Lee, “Pid controller tuning to obtain desired closed-loop responses for cascade control systems,” *IFAC Proceedings Volumes*, vol. 31, no. 11, pp. 613–618, 1998, 5th IFAC Symposium on Dynamics and Control of Process Systems 1998 (DYCOPS 5), Corfu, Greece, 8-10 June 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017449949

[30] J. Gao, Q. Li, Q. Wang, W. Jiang, and R. Kennel, “Servo press drive using model predictive control of motor current,” in *2019 IEEE International*

Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE). IEEE, May 2019.

[31] G. Huang, C. Cui, X. Lei, Q. Li, S. Yan, X. Li, and G. Wang, "A review of optical interferometry for high-precision length measurement," *Micromachines*, vol. 16, no. 1, 2025. [Online]. Available: https://www.mdpi.com/2072-666X/16/1/6

[32] V. Badami and P. DE GROOT, *Displacement Measuring Interferometry*, 02 2013, pp. 157–238.

[33] P. Hariharan, *Basics of Interferometry*, 2nd ed. Amsterdam, Netherlands: Elsevier, 2007.

[34] R. Leach, *Optical Measurement of Surface Topography*, 01 2011.

[35] Y. Lyu, L. Ma, and Q. Yang, "Experimental study on the application of the michelson interferometer," 01 2017.

[36] HyperPhysics, "Michelson Interferometer," http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/michel.html#c1, accessed: July 5, 2025.

[37] V. Badami and P. DE GROOT, *Displacement Measuring Interferometry*, 02 2013, pp. 157–238.

[38] T. Požar, P. Gregorčič, and J. Možina, "Optimization of displacement-measuring quadrature interferometers considering the real properties of optical components," *Appl. Opt.*, vol. 50, no. 9, pp. 1210–1219, Mar 2011. [Online]. Available: https://opg.optica.org/ao/abstract.cfm?URI=ao-50-9-1210

[39] P. Gregorčič, T. Požar, and J. Mozina, "Quadrature phase-shift error analysis using a homodyne laser interferometer," *Optics Express*, vol. 17, pp. 16 322–16 331, 08 2009.

[40] N. Bobroff, "Recent advances in displacement measuring interferometry," *Measurement Science and Technology*, vol. 4, p. 907, 01 1999.

[41] B. Banitalebi, M. Kamarei, M. Karimi, and G. Dadashzadeh, "Impact of the undesired rf-if chain effects on the ls-cma based beamformer in cdma mobile communications," in *2006 IEEE International Conference on Industrial Technology*, 2006, pp. 596–600.

[42] J.-Y. Lee, C.-T. Hsu, S.-H. Chang, and W.-Y. Chen, "Dual beam polarization interferometry for roll angular displacement measurement," *Measurement*, vol. 222, p. 113571, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224123011351

[43] SIOS Meßtechnik GmbH, “Measuring in the electron storage ring,” https://www.sios-precision.com/en/applications/measuring-in-the-electron-storage-ring, accessed: Jul. 6, 2025.

[44] M. Chapman, “Environmental compensation of linear laser interferometer readings,” *Renishaw Technical white paper, TE329*, 2013.

[45] S. Zhang, Y. Lou, Y. Zhou, B. Chen, and L. Yan, “Self-correction of air refractive index in distance measurement using two-color sinusoidal phase modulating interferometry,” *Optics Communications*, vol. 505, p. 127521, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030401821007707

[46] International Organization for Standardization, “ISO 230-2: Test Code for Machine Tools – Part 2: Determination of Accuracy and Repeatability of Positioning of Numerically Controlled Axes,” ISO, Geneva, Switzerland, 2012.

[47] Verein Deutscher Ingenieure (VDI)/VDE, “VDI/VDE 3441: Statistische Prüfung der Arbeits- und Positionsgenauigkeit von Werkzeugmaschinen – Grundlagen,” Verlag des Vereins Deutscher Ingenieure (VDI), Düsseldorf, Germany, Mar. 1977, withdrawn Apr. 15.2022.

[48] M. Gull, “Developments in bsi and is0 machine tool accuracy standards,” *WIT Transactions on Engineering Sciences*, vol. 34, 2025.

[49] B. Mullany, “Evaluation and comparison of the different standards used to define the positional accuracy and repeatability of numerically controlled machining center axes,” *University of North Carolina, Charlotte*, 2007.

[50] S. Tabakovic, M. Zeljkovic, S. Zivanovic, A. Budimir, Z. Dimic, and A. Kosarac, “Calibration of a hybrid machine tool from the point of view of positioning accuracy,” *Applied Sciences*, vol. 14, no. 12, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/12/5275

[51] C. Dey and S. K. Sen, *Industrial automation technologies.* CRC Press, 2020.

[52] T. O. Olawumi, D. W. Chan, S. Ojo, and M. C. Yam, “Automating the modular construction process: A review of digital technologies and future directions with blockchain technology,” *Journal of Building Engineering*, vol. 46, p. 103720, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352710221015783

[53] Pejović Predrag, “Application of python programming language in measurements,” vol. 32, no. 1, pp. 1–23 PT– Article AB– Application of Python programming language in automation of measurement systems and creating virtual instruments is discussed in this paper. Requirements imposed to the software in order to perform these tasks are listed, and Python modules that

support them are presented. Application of proposed techniques are illustrated in seven examples in different application areas. Analysis of software evolution, as well as the evolution of professional education yields conclusion that application of Python in automating measurement systems is promising., 2019.

[54] A. Quenon, E. Daubie, V. Moeyaert, and F. Dualibe, "R-testbench: a python library for instruments remote control and electronic test bench automation," in *Sensors and Electronic Instrumentation Advances: Proceedings of the 6th International Conference on Sensors and Electronic Instrumentation Advances (SEIA'2020) and the 2nd IFSA Frequency & Time Conference (IFTC'2020)*, 2020, pp. 47–50.

[55] 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, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[56] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56 – 61.

[57] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G.-L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, Y. Vázquez-Baeza, and SciPy 1.0 Contributors, "SciPy 1.0: fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020.

[58] J. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, pp. 90–95, 06 2007.

[59] B. D. Hall, "The gum tree calculator: A python package for measurement modelling and data processing with automatic evaluation of uncertainty," *Metrology*, vol. 2, no. 1, pp. 128–149, 2022. [Online]. Available: https://www.mdpi.com/2673-8244/2/1/9

[60] PyVISA Authors. (2025) Visa Library — pyvisa documentation. PyVISA 1.15.1.dev18+g94c0f90. [Online]. Available: https://pyvisa.readthedocs.io/en/latest/api/visalibrarybase.html

[61] O. R. Dubonnet and the Python OPC–UA contributors. (2025) Python OPC–UA Documentation. Latest docs, revision 131ad8ce. [Online]. Available: https://python-opcua.readthedocs.io/en/latest

[62] ACS Motion Control, "Acs motion control," https://acsmotioncontrol.com, 2025, accessed: 2025-07-30.

[63] SIOS Meßtechnik GmbH, "Sios precision measurement technology," https://www.sios-precision.com/en, 2025, accessed: 2025-07-30.

[64] E. Ikonen, J. Kauppinen, T. Korkolainen, J. Luukkainen, and K. Riski, "Interferometric calibration of gauge blocks by using one stabilized laser and a white-light source," *Applied Optics*, vol. 30, no. 31, pp. 4477–4478, 1991.

[65] H.-W. Lee, J.-R. Chen, S.-P. Pan, H.-C. Liou, and P.-E. Hsu, "Relationship between iso 230-2/-6 test results and positioning accuracy of machine tools using lasertracer," *Applied Sciences*, vol. 6, p. 105, 2016.

[66] M. Bojanić, G. Jovičić, and M. Zeljković, "Determination of accuracy of positioning cnc machine tools," *Annals of the Faculty of Engineering Hunedoara*, vol. 13, no. 3, pp. 45–52, 2015.

[67] H. Haitjema, "Calibration of displacement laser interferometer systems for industrial metrology," *Sensors*, vol. 19, p. 4100, 2019.

[68] G. Berkovic and E. Shafir, "Optical methods for distance and displacement measurements," *Advances in Optics and Photonics*, vol. 4, pp. 441–471, 2012.

[69] L. Yan, Y. Yan, B. Chen, and Y. Lou, "A differential phase-modulated interferometer with rotational error compensation for precision displacement measurement," *Applied Sciences*, vol. 12, p. 5002, 2022.

[70] C.-S. Tran, T.-H. Hsieh, and W.-Y. Jywe, "Laser r-test for angular positioning calibration and compensation of five-axis machine tools," *Applied Sciences*, vol. 11, p. 9507, 2021.

[71] F. Ezedine, J.-M. Linares, J. Chaves-Jacob, and J.-M. Sprauel, "Measurement parameters optimised for sequential multilateration in calibrating a machine tool with a doe method," *Applied Sciences*, vol. 6, p. 313, 2016.

[72] C.-H. Yu, P.-C. Tung, Y.-C. Wang, L.-H. Shyu, and E. Manske, "Linear displacement calibration system integrated with a novel auto-alignment module for optical axes," *Sensors*, vol. 20, p. 2462, 2020.

[73] A. Muelaner and P. Calleja, "Traceability of on-machine tool measurement: A review," *Sensors*, vol. 17, p. 1605, 2017.

[74] H. Liu, Y. Guo, J. Liu, and W. Niu, "Dynamic error compensation for ball screw feed drive systems based on prediction model," *Machines*, vol. 13, no. 5, p. 433, 2025.

[75] L. Tao, L. Zhang, X. Wang, X. Fan, and H. Shi, "Dynamic screw lead accuracy measurement: Research on the compensation of the workpiece placed away from the ideal position," *Sensors*, vol. 24, no. 21, p. 6829, 2024.

[76] Q. Xiang, C. Chen, and Y. Jiang, "Vibration suppression of collaborative robot servo with dual encoders," *Applied Sciences*, vol. 15, no. 5, p. 2548, 2025.

[77] Y. Zhou, W. Zong, Q. Tan, Z. Hu, T. Sun, and L. Li, "End effect analysis of a slot-less long-stator permanent magnet linear synchronous motor," *Symmetry*, vol. 13, no. 10, p. 1939, 2021.

[78] H. Yu, G. Zheng, Y. Liu, J. Zhao, G. Wei, and H. Jiang, "Research on the dynamic characteristics of a dual linear-motor differential-drive micro-feed servo system," *Applied Sciences*, vol. 14, no. 8, p. 3170, 2024.

[79] Q. Zhao, M. Qiang, Y. Hou, S. Chen, and T. Lai, "Research developments of aerostatic thrust bearings: A review," *Applied Sciences*, vol. 12, no. 23, p. 11887, 2022.

[80] H.-J. Lee and D. Ahn, "Development of air bearing stage using flexure for yaw motion compensation," *Actuators*, vol. 11, no. 4, p. 100, 2022.

[81] W. Hao, G. Zhang, W. Liu, H. Liu, and Y. Wang, "Methods for reducing cogging force in permanent magnet machines: A review," *Energies*, vol. 16, no. 1, p. 422, 2023.

[82] Q. Wang, B. Zhao, J. Zou, and Y. Li, "Minimization of cogging force in fractional-slot permanent magnet linear motors with double-layer concentrated windings," *Energies*, vol. 9, no. 11, p. 918, 2016.

[83] H. Zhang and B. Kou, "Design and optimization of a lorentz-force-driven planar moor," *Applied Sciences*, vol. 7, no. 1, p. 7, 2017.

[84] P. R. Eckert, A. F. Flores Filho, E. Perondi, J. Ferri, and E. Goltz, "Design methodology of a dual-halbach array linear actuator with thermal-electromagnetic coupling," *Sensors*, vol. 16, no. 3, p. 360, 2016.

[85] Z. Wu, Z. Wu, I.-M. Chen, and Q. Xu, "Recent advances in piezoelectric compliant devices for ultrahigh-precision engineering," *Micromachines*, vol. 15, no. 12, p. 1456, 2024.

[86] G. Hu, W. Xin, M. Zhang, G. Chen, J. Man, and Y. Tian, "Development of a fast positioning platform with a large stroke based on a piezoelectric actuator for precision machining," *Micromachines*, vol. 15, no. 8, p. 1050, 2024.

[87] X. Zeng, F. Nan, T. Li, C. Mo, J. Su, K. Wei, and X. Zhang, "Integrating strain gauge feedback with adaptive sliding mode motion control for piezoelectric nanopositioning stage," *Actuators*, vol. 14, no. 2, p. 79, 2025.

[88] G. Tang, Z. Wang, S. Liu, C. Li, and J. Wang, "Real-time hyperspectral video acquisition with coded slits," *Sensors*, vol. 22, no. 3, p. 822, 2022.

[89] H. Chen, L. Li, R. Li, G. Yu, and Q. Chen, "Design and analysis of a long-stroke and high-precision positioning system for scanning beam interference lithography," *Electronics*, vol. 12, no. 24, p. 4960, 2023.

[90] Renishaw, "Xl-80 laser system: Technical specification," https://www.renishaw.com, 2024, wotton-under-Edge, UK.

[91] K. Wang, G. Xing, P. Yang, M. Wang, Z. Wang, and Q. Tian, "High-bandwidth heterodyne laser interferometer for the measurement of high-intensity focused ultrasound pressure," *Micromachines*, vol. 14, no. 12, p. 2225, 2023.

[92] Y. Qian, J. Li, Q. Feng, Q. He, and F. Long, "Error analysis of heterodyne interferometry based on one single-mode polarization-maintaining fiber," *Sensors*, vol. 23, no. 8, p. 4108, 2023.

[93] H. Kang, J. Lee, Y.-J. Kim, and S.-W. Kim, "Phase-locked synthetic wavelength interferometer using a femtosecond laser for absolute distance measurement without cyclic error," *Sensors*, vol. 23, no. 14, p. 6253, 2023.

[94] B. Chang, T. Tan, J. Du *et al.*, "Dispersive fourier transform based dual-comb ranging," *Nature Communications*, vol. 15, p. 4990, 2024.

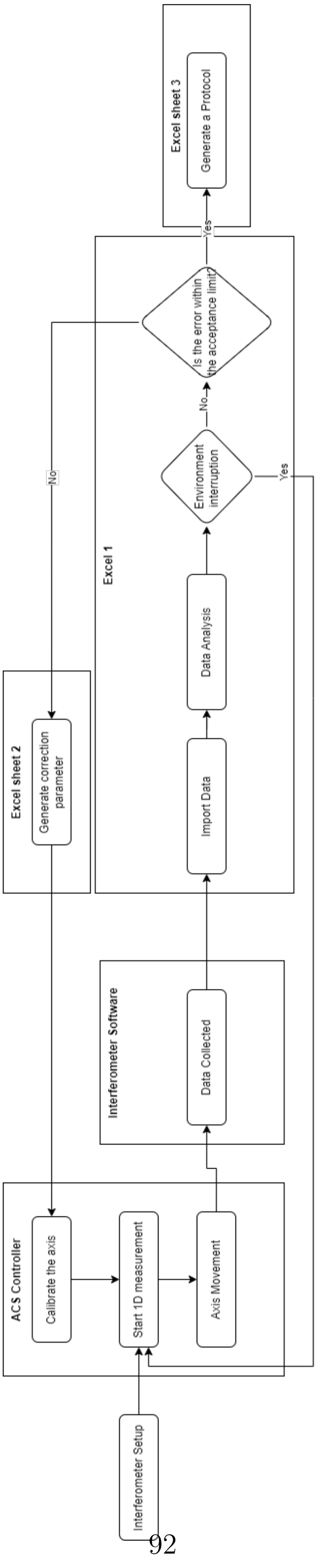
[95] J. Zhi, M. Xu, Y. Liu, M. Wang, C. Shao, and H. Wu, "Ultrafast and precise distance measurement via real-time chirped pulse interferometry," *Advanced Photonics Nexus*, vol. 4, no. 2, p. 026012, 2025.

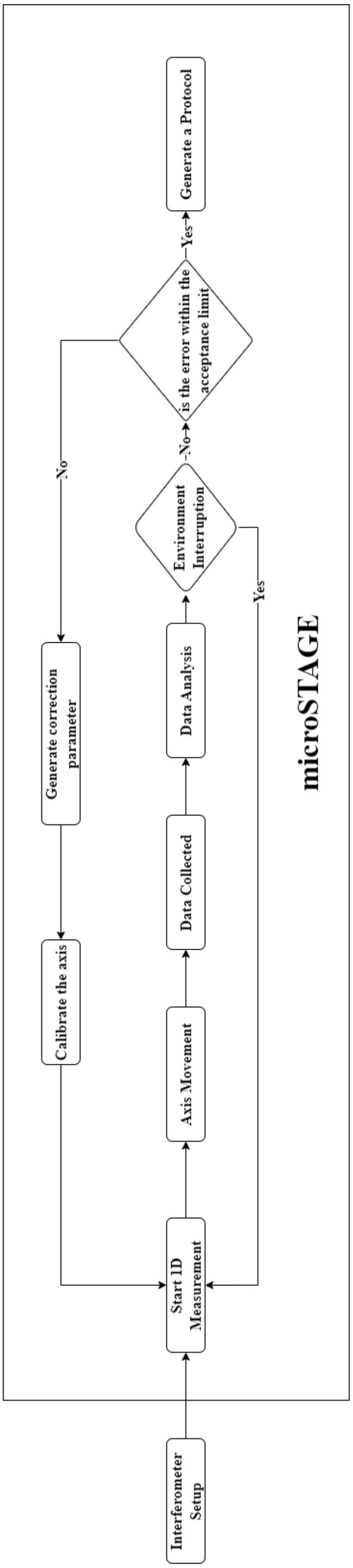
Bibliography from TUC

[TUC1] T.-D. Tran, H. Schlegel, and R. Neugebauer, “Application of modal analysis for commissioning of drives on machine tools,” in *Procedia CIRP*, vol. 40, 2016, pp. 115–120.

[TUC2] F. Abouzid, S. Saleh, S. Harras, W. Hardt, and H. Ghennioui, “Evaluation of yolov7 deployment performance on various platforms (cpus, gpus) and strategies,” *Embedded Selforganising Systems*, vol. 10, no. 8, 2023. [Online]. Available: https://doi.org/10.14464/ess.v10i8.660

A. WorkFlow diagrams




microSTAGE
Interferometer Setup
Start 1D Measurement
Axis Movement
Data Collected
Data Analysis
Environment Interruption
Yes
No
is the error within the acceptance limit
Yes
Generate a Protocol
No
Generate correction parameter
Calibrate the axis

B. Sample Protocol Reports before calibration

1D Measurement Analysis

Measurement Information

Machine Type:	
Machine Number:	
Customer:	TU-Chemnitz
Operator:	Alhassan Khalil
Creation Date:	Wednesday, August 6, 2025

Axis Information

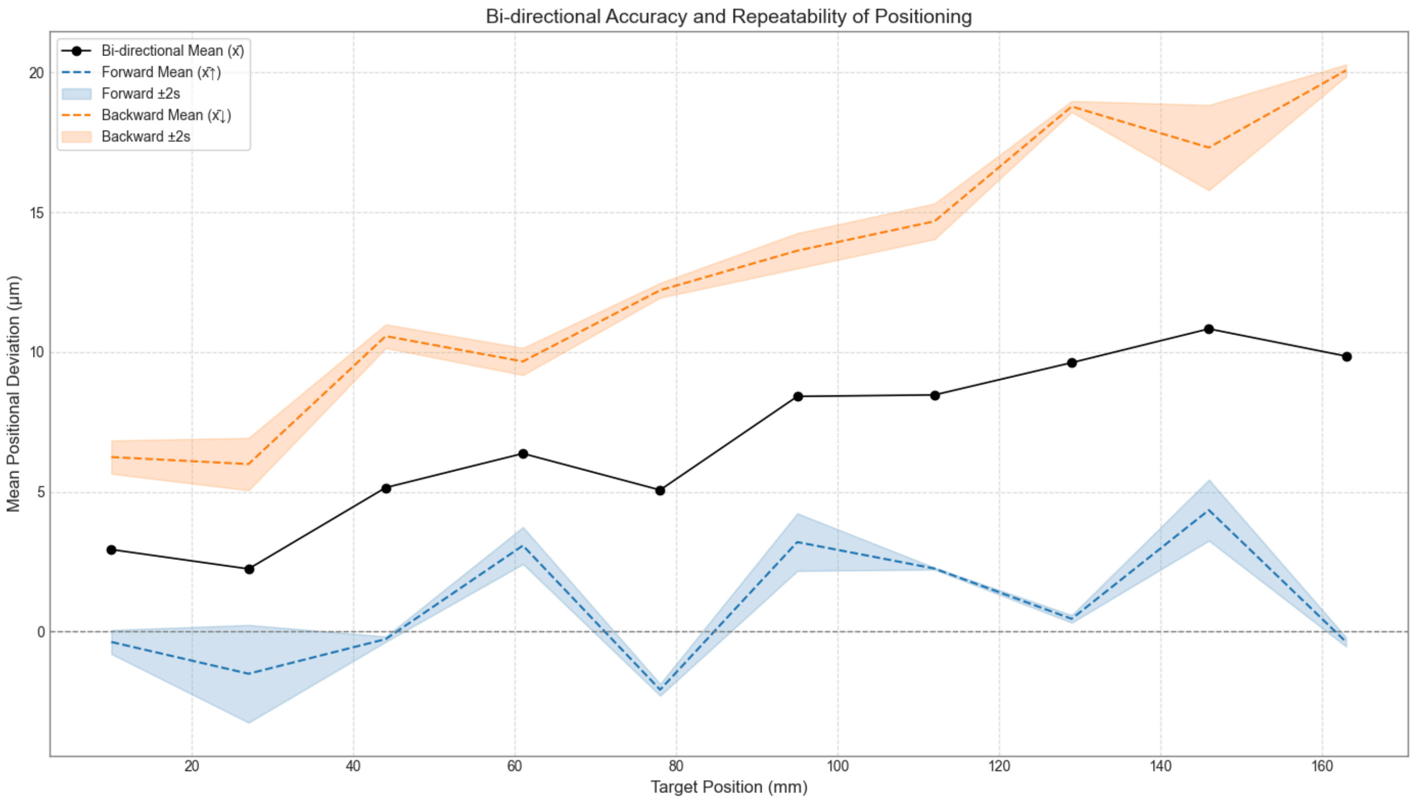
Manufacturer:	AeroTech
Type:	PRO165L
Serial Number:	132321

Measurement Method

Measurement Point Sequence:	Pendulum
Type of Execution:	Bidirectional
Number of Measurement Point:	10
Number of Cycle:	5
First Point:	10.000 mm
Last Point:	163.000 mm
Measurement Point [mm]:	10,27,44,61,78,95,112,129,146,163

Measurement Analsis

Measurement Plot



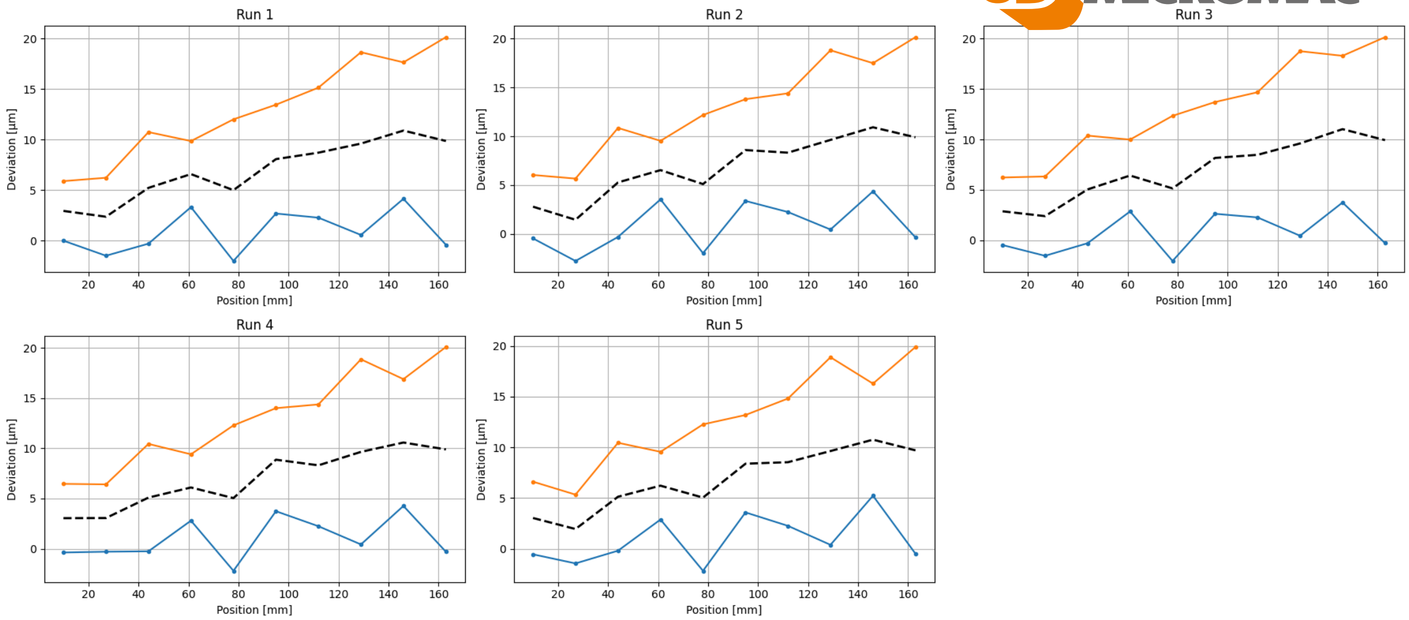
Measurement Data

Accuracy (A):	23.53 µm
Forward Accuracy (A↑):	8.68 µm
Backward Accuracy (A↓):	15.22 µm
Systematic positional deviation (E):	22.15 µm
Forward E↑:	6.42 µm
Backward E↓:	14.08 µm
Repeatability (R):	20.81 µm
Forward Repeatability (R↑):	3.49 µm
Backward Repeatability (R↓) :	3.04 µm
Mean reversal (B):	-12.04 µm

Environment Data

Max Temperature:	21.14 °C
Min Temperature:	21.13 °C
ΔT:	0.01 °C
ΔT Ref Temp:	0.14 °C
Max Air Pressure:	98612.00 Pa
Min Air Pressure:	98605.00 Pa

Individual Runs Plot



1D Measurement Analysis

Measurement Information

Machine Type:	
Machine Number:	
Customer:	TU-Chemnitz
Operator:	Alhassan Khalil
Creation Date:	Wednesday, August 6, 2025

Axis Information

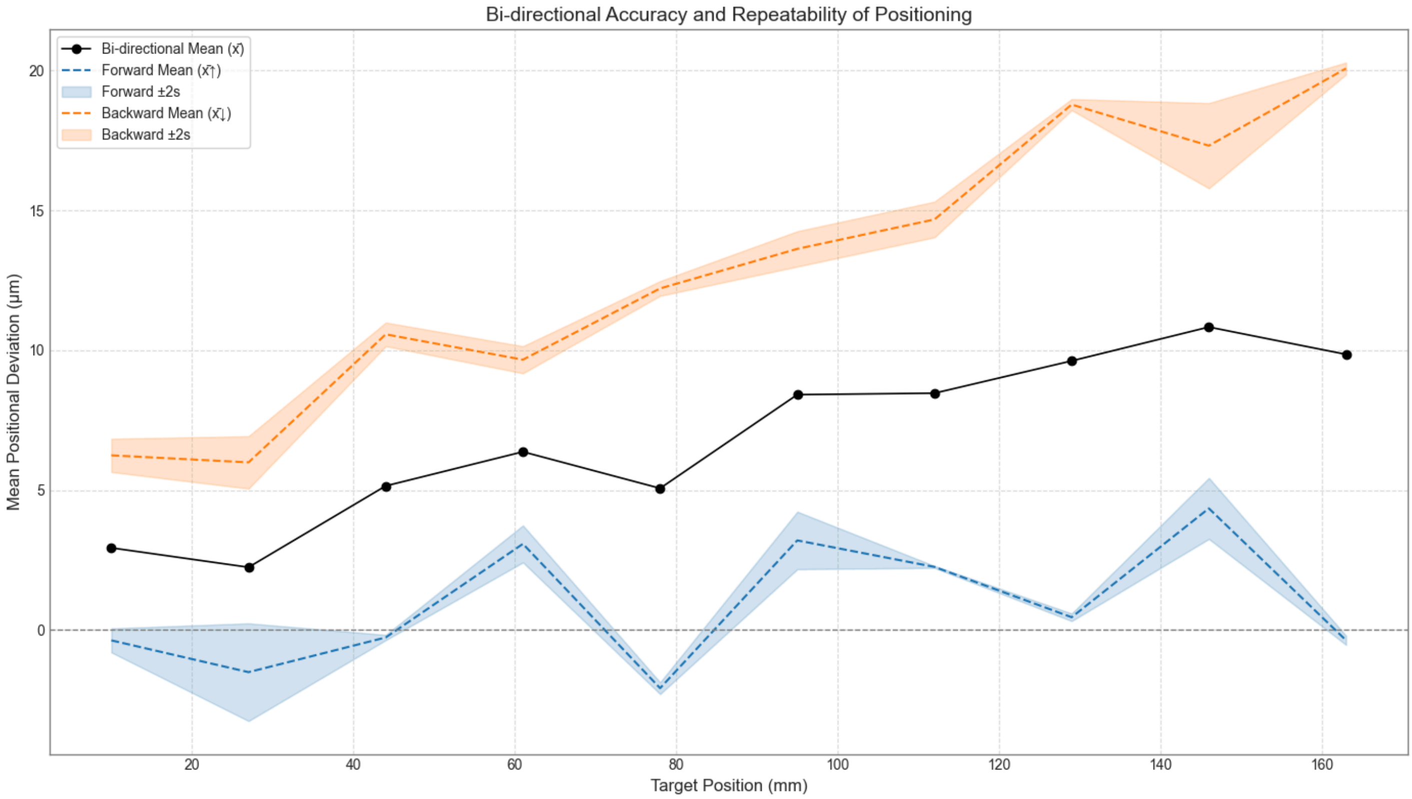
Manufacturer:	AeroTech
Type:	PRO165L
Serial Number:	132321

Measurement Method

Measurement Point Sequence:	Pendulum
Type of Execution:	Bidirectional
Number of Measurement Point:	10
Number of Cycle:	5
First Point:	10.000 mm
Last Point:	163.000 mm
Measurement Point [mm]:	10,27,44,61,78,95,112,129,146,163

Measurement Analsis

Measurement Plot



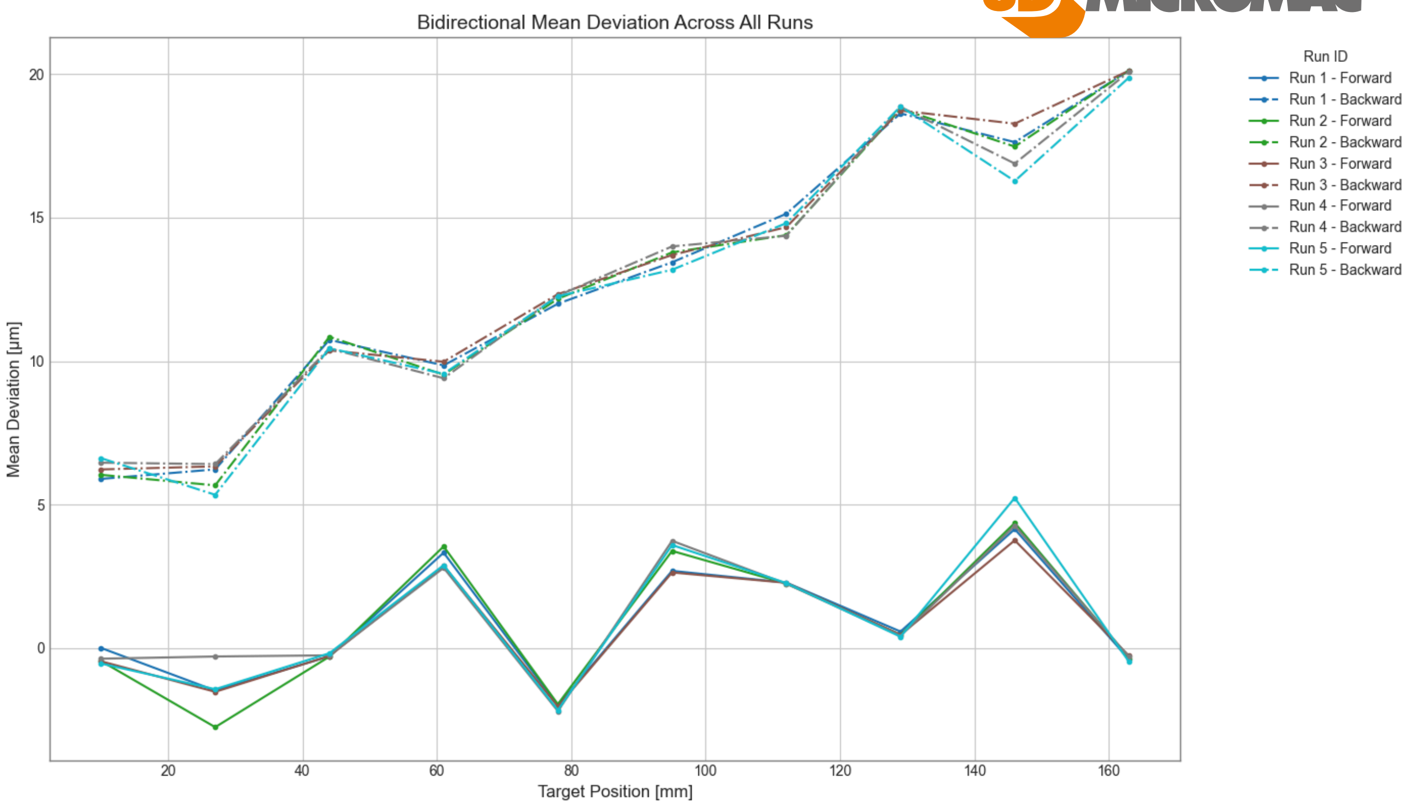
Measurement Data

Accuracy (A):	23.53 µm
Forward Accuracy (A↑):	8.68 µm
Backward Accuracy (A↓):	15.22 µm
Systematic positional deviation (E):	22.15 µm
Forward E↑:	6.42 µm
Backward E↓:	14.08 µm
Repeatability (R):	20.81 µm
Forward Repeatability (R↑):	3.49 µm
Backward Repeatability (R↓) :	3.04 µm
Mean reversal (B):	-12.04 µm

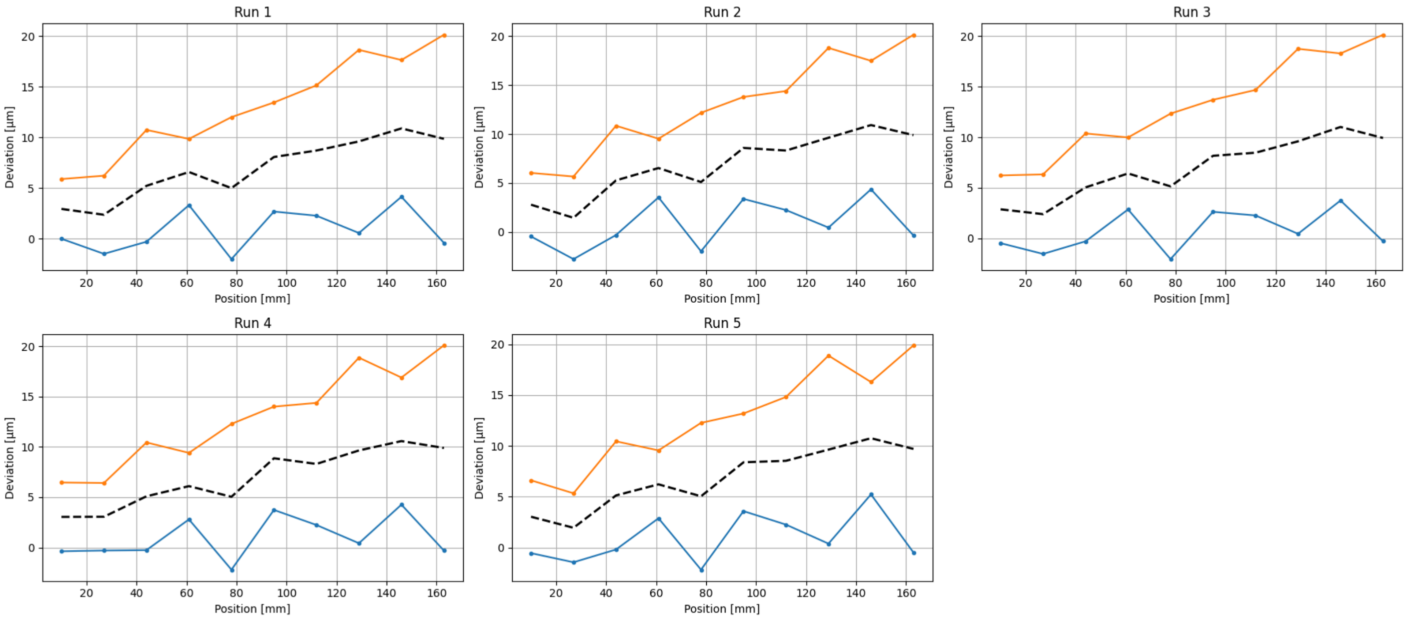
Environment Data

Max Temperature:	21.14 °C
Min Temperature:	21.13 °C
ΔT:	0.01 °C
ΔT Ref Temp:	0.14 °C
Max Air Pressure:	98612.00 Pa
Min Air Pressure:	98605.00 Pa

Backward and forward Runs Plot



Individual Runs Plot



C. Sample Protocol Reports after calibration

1D Measurement Analysis

Measurement Information

Machine Type:	Stage Calibration
Machine Number:	169
Customer:	Tu-Chemnitz
Operator:	Alhassan
Creation Date:	Monday, June 16, 2025

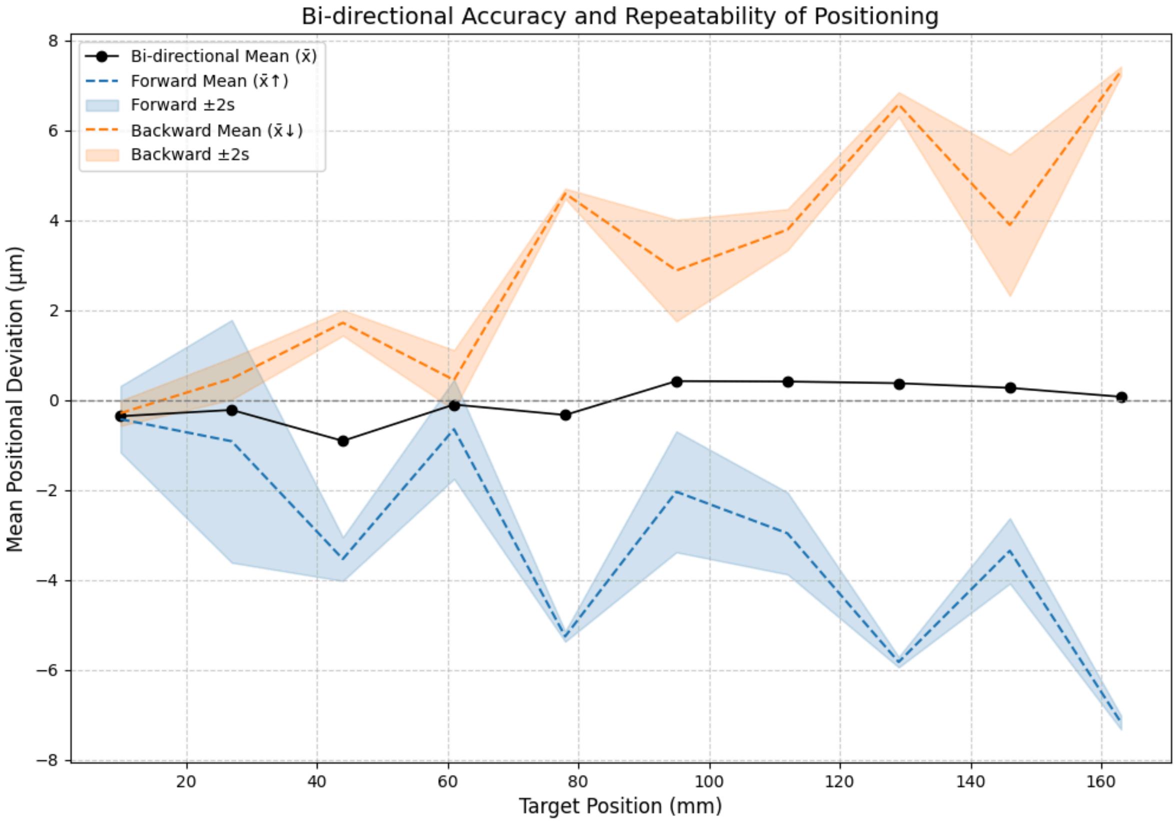
Axis Information

Manufacturer:	AeroTech
Type:	165Pro
Serial Number:	123321

Measurement Method

Measurement Point Sequence:	Linear
Type of Execution:	Bidirectional
Number of Measurement Point:	10
Number of Cycle:	5
First Point:	10.000 mm
Last Point:	163.000 mm
Measurement Point [mm]:	10,27,44,61,78,95,112,129,146,163

Measurement Plot



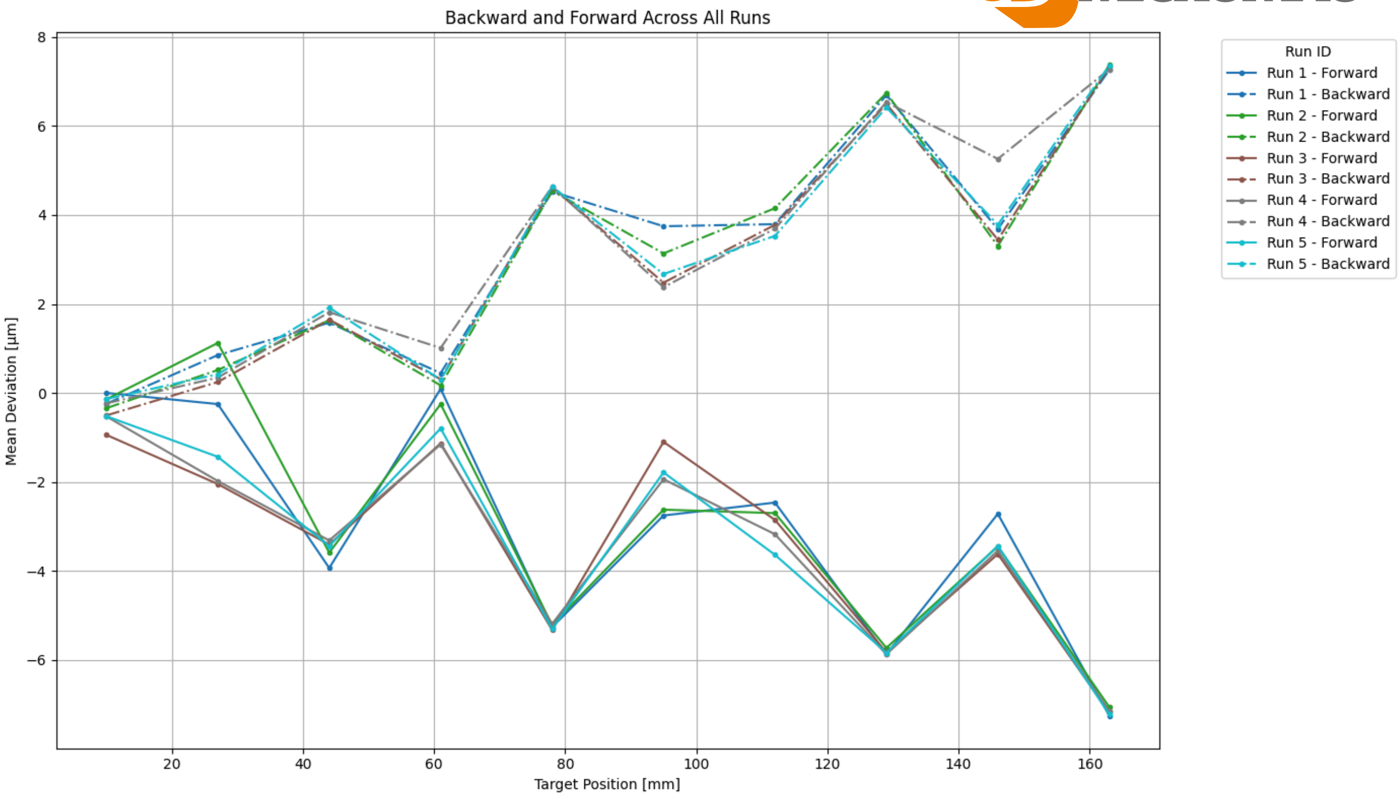
Measurement Data

Accuracy (A):	14.74 µm
Forward Accuracy (A↑):	9.11 µm
Backward Accuracy (A↓):	7.99 µm
Systematic positional deviation (E):	14.49 µm
Forward E↑:	6.75 µm
Backward E↓:	7.61 µm
Repeatability (R):	14.74 µm
Forward Repeatability (R↑):	5.40 µm
Backward Repeatability (R↓) :	3.15 µm
Mean reversal (B):	-6.36 µm

Environment Data

Max Temperature:	21.16 °C
Min Temperature:	21.15 °C
ΔT:	0.01 °C
ΔT Ref Temp:	0.15 °C
Max Air Pressure:	98608.00 Pa
Min Air Pressure:	98604.00 Pa

Backward and forward Runs Plot



Chemnitzer Informatik-Berichte

In der Reihe der Chemnitzer Informatik-Berichte sind folgende Berichte erschienen:

CSR-23-01 Stephan Lede, René Schmidt, Wolfram Hardt, Analyse des Ressourcenverbrauchs von Deep Learning Methoden zur Einschlagslokalisierung auf eingebetteten Systemen, Januar 2023, Chemnitz

CSR-23-02 André Böhle, René Schmidt, Wolfram Hardt, Schnittstelle zur Datenakquise von Daten des Lernmanagementsystems unter Berücksichtigung bestehender Datenschutzrichtlinien, Januar 2023, Chemnitz

CSR-23-03 Falk Zaumseil, Sabrina, Bräuer, Thomas L. Milani, Guido Brunnett, Gender Dissimilarities in Body Gait Kinematics at Different Speeds, März 2023, Chemnitz

CSR-23-04 Tom Uhlmann, Sabrina Bräuer, Falk Zaumseil, Guido Brunnett, A Novel Inexpensive Camera-based Photoelectric Barrier System for Accurate Flying Sprint Time Measurement, März 2023, Chemnitz

CSR-23-05 Samer Salamah, Guido Brunnett, Sabrina Bräuer, Tom Uhlmann, Oliver Rehren, Katharina Jahn, Thomas L. Milani, Güunter Daniel Rey, NaturalWalk: An Anatomy-based Synthesizer for Human Walking Motions, März 2023, Chemnitz

CSR-24-01 Seyhmus Akaslan, Ariane Heller, Wolfram Hardt, Hardware-Supported Test Environment Analysis for CAN Message Communication, Juni 2024, Chemnitz

CSR-24-02 S. M. Rizwanur Rahman, Wolfram Hardt, Image Classification for Drone Propeller Inspection using Deep Learning, August 2024, Chemnitz

CSR-24-03 Sebastian Pettke, Wolfram Hardt, Ariane Heller, Comparison of maximum weight clique algorithms, August 2024, Chemnitz

CSR-24-04 Md Shoriful Islam, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Predictive Learning Analytics System, August 2024, Chemnitz

CSR-24-05 Sopuluchukwu Divine Obi, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Frontend for Agents in a Virtual Tutoring System, August 2024, Chemnitz

CSR-24-06 Saddaf Afrin Khan, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Diagnostic Learning Analytics System, August 2024, Chemnitz

Chemnitzer Informatik-Berichte

CSR-24-07 Túlio Gomes Pereira, Wolfram Hardt, Ariane Heller, Development of a Material Classification Model for Multispectral LiDAR Data, August 2024, Chemnitz

CSR-24-08 Sumanth Anugandula, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Virtual Agent for Interactive Learning Scenarios, September 2024, Chemnitz

CSR-25-01 Md. Ali Awlad, Hasan Saadi Jaber Aljzaere, Wolfram Hardt, AUTOSAR Software Component for Atomic Straight Driving Patterns, März 2025, Chemnitz

CSR-25-02 Billava Vasantha Monisha, Hasan Saadi Jaber Aljzaere, Wolfram Hardt, Automotive Software Component for QT Based Car Status Visualization, März 2025, Chemnitz

CSR-25-03 Zahra Khadivi, Batbayar Battseren, Wolfram Hardt, Acoustic-Based MAV Propeller Inspection, Mai 2025, Chemnitz

CSR-25-04 Tripti Kumari Shukla, Ummay Ubaida Shegupta, Wolfram Hardt, Time Management Tool Development to Support Self-regulated Learning, August 2025, Chemnitz

CSR-25-05 Ambu Babu, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Retrieval Model based Backend of a Tutoring Agent, August 2025, Chemnitz

CSR-25-06 Shahid Ismail, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Generative Model based Backend of Tutoring Agent, August 2025, Chemnitz

CSR-25-07 Chaitanya Sravanthi Akula, Ummay Ubaida Shegupta, Wolfram Hardt, Integration of Learning Analytics into the ARC-Tutoring Workbench, August 2025, Chemnitz

CSR-25-08 Jörn Roth, Reda Harradi, Wolfram Hardt, Implementation of a Path Planning Algorithm for UAV Navigation, Dezember 2025, Chemnitz

CSR-25-09 Alhassan Khalil, Reda Harradi, Stephan Rupf, Wolfram Hardt, Development of an Automation Framework for 1D Measurement, Dezember 2025, Chemnitz

Chemnitzer Informatik-Berichte

ISSN 0947-5125

Herausgeber: Fakultät für Informatik, TU Chemnitz
Straße der Nationen 62, D-09111 Chemnitz