© May 2026 | IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

Design and Development of a Sensor Calibration and Error Analysis Platform

Prof. Vijaykumar Bhanuse[1], Shriyash Jadhav[2], Prajwal Kale[3], Mithil Ambhure[4], Tushar Khurale[5]
[1,2,3,4,5]*Department of Instrumentation and Control Engineering (ICE), Vishwakarma Institute of Technology, Pune, Maharashtra, India*

***Abstract*—Accurate sensor calibration is a fundamental prerequisite for reliable measurement in industrial instrumentation, process control, and quality assurance systems. This paper presents the design and development of a comprehensive hardware and software platform for sensor calibration and error analysis, targeting final-year engineering education and small-scale industrial calibration laboratories. The platform integrates an Arduino/ESP32-based data acquisition subsystem with analog signal conditioning circuits to collect raw sensor outputs from temperature, pressure, and load sensors. Collected data is processed using Python-based computational tools implementing linear regression, curve fitting, and statistical error analysis algorithms. The system computes and reports calibration curves, sensitivity coefficients, linearity errors, hysteresis errors, repeatability errors, and offset corrections. Compensation equations derived from the calibration process are stored and applied to correct subsequent sensor readings in real time. Experimental evaluation using the LM35 temperature sensor demonstrates a maximum calibration error reduction from ±2.5°C uncalibrated to ±0.3°C post-calibration. The platform is designed in alignment with ISO 17025 calibration laboratory standards and provides a practical, cost-effective tool for demonstrating industrial instrumentation concepts. Results confirm that systematic calibration with rigorous error analysis significantly improves sensor measurement accuracy, making the platform suitable for process industries, pharmaceutical applications, and educational instrumentation laboratories.**

***Index Terms*—Sensor Calibration, Error Analysis, Signal Conditioning, Linear Regression, Hysteresis, Repeatability, Linearity, Instrumentation, ISO 17025, Compensation Equation**

I. INTRODUCTION

Accurate measurement is the cornerstone of reliable industrial process control, quality assurance, and safety-critical instrumentation. In any measurement system, a sensor translates a physical quantity into an electrical signal; however, due to manufacturing tolerances, environmental influences, aging, and non-linear transduction characteristics, the raw output of a sensor rarely represents the true value of the measurand with sufficient accuracy for industrial applications. Sensor calibration the process of establishing a defined relationship between the sensor output and the true value of the measurand under specified conditions is therefore essential to ensure measurement traceability and accuracy [1].

Industrial process industries including oil and gas, pharmaceutical manufacturing, food processing, and power generation rely on precise measurements of temperature, pressure, flow, and force for safe and efficient operation. Sensors deployed in these environments are subject to drift, thermal effects, electromagnetic interference, and mechanical stress that alter their transfer characteristics over time. Without periodic calibration and error compensation, measurement errors can accumulate to levels that compromise product quality, violate safety regulations, and lead to costly production failures [2].

The international standard ISO 17025, which governs the competence of testing and calibration laboratories, mandates that all measuring instruments used in calibration processes be traceable to national measurement standards and that calibration uncertainty be documented and communicated. Despite the critical importance of calibration, many small-scale industrial operations and educational institutions lack access to specialized calibration equipment, software, and trained personnel. This gap motivates the development of an accessible, low-cost calibration platform that embodies the methodological rigor of professional calibration practice [3].

This paper presents the design and development of a sensor calibration and error analysis platform that integrates hardware-level data acquisition using a microcontroller-based system with software-level mathematical analysis implemented in Python. The platform supports calibration of multiple sensor types including temperature (LM35, PT100), pressure (MPX5100), and force (load cell) sensors, and performs comprehensive error analysis encompassing linearity, hysteresis, repeatability, and sensitivity assessment. Calibration curves are generated through linear regression and polynomial curve fitting, and compensation equations are derived and applied for real-time error correction.

The key contributions of this work are: (a) an integrated, low-cost calibration platform using commercially available microcontroller hardware; (b) a Python-based analysis pipeline implementing multiple error analysis techniques compliant with ISO 17025 methodology; (c) visualization tools for calibration curves, error plots, and hysteresis diagrams; (d) a compensation equation framework for real-time sensor output correction; and (e) experimental validation demonstrating significant accuracy improvement post-calibration.

II. LITERATURE SURVEY

The field of sensor calibration and error analysis has been extensively studied across multiple application domains. Huijsing et al. [4] provided a comprehensive treatment of precision instrumentation amplifiers and signal conditioning circuits, establishing fundamental principles for low-noise acquisition of sensor outputs. Their work demonstrated that careful attention to impedance matching, offset compensation, and gain stability is critical for achieving measurement accuracies in the millivolt range required for thermocouple and RTD calibration.

In the domain of temperature sensor calibration, Nicholas and White [5] systematically described calibration procedures for platinum resistance thermometers, thermocouples, and thermistors against international temperature scale reference points. Their methodology, now incorporated into ITS-90, forms the basis for traceable temperature calibration and directly informs the RTD calibration procedure implemented in the present platform.

Regarding error analysis methodologies, the Joint Committee for Guides in Metrology (JCGM) Guide to the Expression of Uncertainty in Measurement (GUM) [6] provides the internationally recognized framework for identifying, quantifying, and propagating measurement uncertainties. The GUM approach distinguishes between Type A uncertainties evaluated by statistical analysis of repeated measurements and Type B uncertainties evaluated from other sources such as manufacturer specifications and reference standard calibration certificates.

Several microcontroller-based calibration systems have been reported in recent literature. Ahmed et al. [7] developed an Arduino-based pressure sensor calibration system for industrial applications, achieving a repeatability error below 0.15% of full scale. Kumar and Singh [8] demonstrated a low-cost temperature calibration system using Raspberry Pi for pharmaceutical cold-chain monitoring, validating against NIST-traceable reference thermometers. These works confirm the viability of microcontroller-based platforms for practical calibration applications but do not address multi-sensor, multi-error-type analysis within a unified software environment, which the present work addresses.

In the area of software-based calibration tools, National Instruments LabVIEW has been widely used for laboratory-grade calibration applications [9]. However, the high licensing cost and hardware dependency of LabVIEW make it inaccessible for educational and resource-constrained environments. The present work addresses this limitation by employing the open-source Python ecosystem, which provides equivalent mathematical and visualization capabilities at zero licensing cost.

III. METHODOLOGY

The methodology adopted in this work follows a structured eight-stage calibration process grounded in ISO 17025 requirements and established instrumentation practice. Each stage is described below.

A. Sensor Data Acquisition

Sensor outputs are acquired using an Arduino Uno or ESP32 microcontroller equipped with a 10-bit

(Arduino) or 12-bit (ESP32) analog-to-digital converter. Each sensor is interfaced through a purpose-built signal conditioning circuit that performs amplification, filtering, and level shifting to map the sensor output range to the ADC input range. The digitized readings are transmitted serially at 9600 baud to the host computer for logging and analysis. Acquisition is performed at a configurable sampling rate of 1–10 Hz. For each calibration point, a minimum of 30 successive readings is collected and averaged to reduce random noise effects.

B. Reference Value Acquisition

Reference values are acquired using calibrated reference instruments traceable to national standards. For temperature calibration, a NIST-traceable reference thermometer with ±0.05°C accuracy serves as the reference. For pressure calibration, a precision deadweight tester or calibrated digital pressure gauge with ±0.025% full-scale accuracy is used. Reference values are recorded simultaneously with sensor readings at each calibration point throughout the measurement range, following an ascending and descending sequence to enable hysteresis characterization.

C. Calibration Data Logging

Sensor outputs and corresponding reference values are logged in a structured CSV format using a Python data logging script that timestamps each entry and computes running statistics (mean, standard deviation, range) for quality monitoring. The calibration dataset for each sensor includes at least five equidistant calibration points spanning the full rated measurement range, with ascending and descending sequences for hysteresis analysis and three independent repetitions for repeatability assessment.

D. Calibration Curve Fitting

Calibration curves are generated by fitting mathematical models to the dataset of (reference value, sensor output) pairs. For sensors with nominally linear responses (LM35, PT100 over limited ranges), first-order linear regression yields the calibration equation:

$$Y = mX + c$$

where Y is the corrected measurement value, X is the raw sensor output, m is the slope (sensitivity), and c is the intercept (offset). For sensors with non-linear characteristics (MPX5100 pressure sensor, load cell), second-order polynomial regression is applied:

$$Y = aX^2 + bX + c$$

Curve fitting is implemented using the NumPy polyfit function, which employs the least-squares method to minimize the sum of squared residuals. The goodness of fit is assessed using the coefficient of determination R^2, with $R^2 > 0.999$ required for acceptance as a valid calibration model.

E. Error Analysis

Comprehensive error analysis is performed encompassing five error metrics as detailed in Section VIII.

F. Compensation Equation Development

The inverse of the calibration curve constitutes the compensation equation, which is applied to convert raw sensor outputs to calibrated measurement values in real time. For a linear calibration model Y = mX + c, the compensation equation is: X_calibrated = (Y_raw – c) / m. The compensation equation parameters are stored in a configuration file and applied automatically to all subsequent sensor readings processed through the platform.

IV. SYSTEM ARCHITECTURE

The proposed calibration platform follows a three-tier architecture comprising a sensor interface layer, a data acquisition and processing layer, and a visualization and reporting layer. Fig. 1 illustrates the overall system architecture.

[System Architecture Diagram]

Sensors (LM35 / PT100 / MPX5100 / Load Cell)

↓ Analog Output

Signal Conditioning Circuit (Amplification + Filtering)

↓ Conditioned Analog Signal

Arduino/ESP32 ADC (10/12-bit Conversion)

↓ Serial Data Stream (9600 baud)

Python Processing Layer (Pandas + NumPy + Scikit-learn)

↓ Calibration Curves + Error Metrics + Compensation Eq.

Visualization Layer (Matplotlib Dashboard + Reports)

Fig. 1. Three-tier system architecture of the sensor calibration and error analysis platform.

A. Sensor Interface Layer

Each sensor is interfaced through a dedicated signal conditioning circuit. The LM35 temperature sensor outputs 10 mV/°C and is connected directly to the ADC input through a unity-gain buffer amplifier to prevent loading effects. The PT100 RTD is interfaced using a Wheatstone bridge configuration with an instrumentation amplifier (INA128) providing a gain of 100, converting the bridge imbalance voltage to an ADC-compatible 0–3.3 V range. The MPX5100 pressure sensor provides a ratiometric output of 0.2–4.7 V across its 0–100 kPa range and is connected directly to the ADC. The load cell output is amplified using an HX711 24-bit ADC module that provides dedicated load cell signal conditioning, eliminating the need for an external amplifier circuit.

B. Data Acquisition Layer

The Arduino Uno and ESP32 serve as the primary data acquisition units. The Arduino Uno provides a 10-bit ADC with ±2 LSB integral non-linearity, while the ESP32 provides a 12-bit ADC (4096 levels) with an input range of 0–3.3 V. For temperature measurements using the LM35 with the Arduino operating at 5 V reference, the ADC resolution corresponds to approximately 0.49°C per LSB (5000 mV ÷ 1024 ÷ 10 mV/°C). The ESP32 provides a finer resolution of approximately 0.08°C per LSB (3300 mV ÷ 4096 ÷ 10 mV/°C), making it preferable for precision temperature calibration applications.

C. Processing and Visualization Layer

The Python-based processing layer ingests the serial data stream, performs statistical preprocessing, implements curve fitting and error analysis algorithms, and generates calibration reports. Pandas provides data manipulation and CSV management. NumPy supports matrix operations for linear algebra computations underlying least-squares regression. Matplotlib generates publication-quality calibration curve plots, error bar diagrams, and hysteresis loop visualizations. The output of the processing layer includes a calibration certificate document (PDF), a set of calibration curve figures, and the compensation equation parameter file.

V. CALIBRATION PROCEDURE

The calibration procedure follows the ascending-descending method recommended by industrial calibration standards. Fig. 2 illustrates the calibration setup for temperature sensor calibration.

[Calibration Setup Diagram]

Reference Thermometer + LM35 Sensor → Temperature Bath

LM35 Output → Signal Conditioning → Arduino ADC → Python Logger

Fig. 2. Calibration setup for LM35 temperature sensor calibration using a temperature bath and NIST-traceable reference thermometer.

The temperature calibration procedure involves the following steps: First, the sensor and reference thermometer are placed in thermal equilibrium within a calibrated temperature bath. Second, the bath temperature is set to the lowest calibration point (0°C) and held steady until thermal equilibrium is confirmed by stable reference thermometer readings (variation < 0.05°Cover 60 seconds). Third, 30 consecutive ADC readings are acquired from the LM35 and averaged. Fourth, the reference temperature is recorded simultaneously. Fifth, steps two through four are repeated for ascending calibration points at 0°C, 25°C, 50°C, 75°C, and 100°C. Sixth, the procedure is repeated in the descending direction (100°C to 0°C) to acquire data for hysteresis analysis. Seventh, the entire ascending-descending sequence is repeated three times for repeatability assessment.

Table I presents the nominal sensor specifications used in the calibration platform. Table II presents a sample calibration dataset for the LM35 temperature sensor.

Table I: Sensor Specifications

Sensor	Range	Sensitivity	Accuracy	Output
LM35	0–150°C	10 mV/°C	±0.5°C	Analog
PT100 RTD	−200–+600°C	0.385 Ω/°C	±0.1°C	Resistive
MPX5100	0–100 kPa	45.9 mV/kPa	±2.5%	Analog
Load Cell	0–50 kg	2 mV/V	±0.05%	Differential

Table II: LM35 Temperature Sensor Calibration Readings (Ascending)

Point	Ref. Temp. (°C)	Sensor Out (mV)	Calc. Temp. (°C)	Abs. Error (°C)	Std. Dev.
1	0.00	10.2	1.02	–1.02	0.12
2	25.00	260.5	26.05	+1.05	0.09
3	50.00	510.8	51.08	+1.08	0.11
4	75.00	758.3	75.83	+0.83	0.14
5	100.00	1008.1	100.81	+0.81	0.10

VI. ERROR ANALYSIS TECHNIQUES

A comprehensive error analysis is essential to quantify the performance characteristics of any measurement instrument. The platform implements five standard error metrics as described below.

A. Sensitivity

Sensitivity is defined as the ratio of the change in the sensor output signal to the corresponding change in the measured variable. For the LM35 sensor, the nominal sensitivity is 10 mV/°C. The platform computes the actual sensitivity from the calibration data as the slope of the best-fit line through the calibration dataset:

$S = \Delta V_{out} / \Delta T = m$

where m is the slope of the linear regression. Deviations of the actual sensitivity from the nominal value indicate gain error that is corrected by the calibration curve. The LM35 calibration experiment yielded an actual sensitivity of 9.98 mV/°C, a deviation of –0.2% from the nominal 10 mV/°C.

B. Linearity Error

Linearity error quantifies the maximum deviation of the actual sensor transfer function from the ideal straight line connecting the endpoint values over the full measurement range. It is expressed as a percentage of the full-scale output (FSO):

$E_{linearity} = (\Delta V_{max} / FSO) \times 100\%$

where ΔV_{max} is the maximum deviation from the best-fit straight line and FSO is the full-scale output span. The LM35 exhibited a linearity error of 0.42% FSO across the 0–100°C calibration range. Fig. 3 illustrates the calibration curve and linearity deviation for the LM35 sensor.

[Calibration Curve: Reference Temperature vs. Sensor Output with Best-Fit Line]

X-axis: Reference Temperature (°C) Y-axis: Sensor Output (mV)

Data points: (0, 10.2), (25, 260.5), (50, 510.8), (75, 758.3), (100, 1008.1)

Best-fit line: $Y = 9.98X + 10.14$ $R^2 = 0.9999$

Fig. 3. LM35 temperature sensor calibration curve showing measured data points, best-fit regression line, and deviation envelope. $Y = 9.98X + 10.14$, $R^2 = 0.9999$.

C. Hysteresis Error

Hysteresis error arises from the difference in sensor output between ascending and descending calibration sequences at the same applied input, and is caused by mechanical friction, residual magnetism, thermal lag, and elastic deformation. Hysteresis is expressed as a percentage of FSO:

$E_{hysteresis} = (V_{ascending} – V_{descending})_{max} / FSO \times 100\%$

For the LM35 sensor operating in a controlled temperature bath, hysteresis was negligible (< 0.05% FSO), as expected for a solid-state semiconductor sensor with no mechanical moving parts. For the pressure sensor and load cell, hysteresis values of 0.18% FSO and 0.12% FSO were recorded, respectively, attributed to viscoelastic effects in the diaphragm and mechanical flexure elements.

D. Repeatability Error

Repeatability is defined as the closeness of agreement between independent results of measurements of the same measurand carried out under identical conditions. It is quantified by the standard deviation of repeated measurements at each calibration point. The repeatability error as a percentage of FSO is:

$E_{repeatability} = (2\sigma / FSO) \times 100\%$

where σ is the standard deviation of repeated readings at a given calibration point across three independent calibration runs. The LM35 exhibited a maximum repeatability error of 0.24% FSO at the 50°C calibration point. The load cell exhibited the highest repeatability error among tested sensors (0.31% FSO) due to sensitivity to mechanical hysteresis and mounting variability.

E. Offset Error

Offset error, also termed zero error or bias, is the constant difference between the sensor output and the true value at the zero-reference point. For the LM35, the measured output at 0°C was 10.2 mV rather than the nominal 0 mV, yielding a zero-point calculated temperature of 1.02°C instead of 0°C. The offset error is captured by the intercept term c of the calibration equation and is automatically corrected by applying the compensation model. Offset errors can arise from manufacturing tolerances in the sensor transducer element, input bias currents in the signal conditioning amplifier, and analog reference voltage inaccuracies in the ADC.

VII. EXPERIMENTAL SETUP

The experimental setup comprises the hardware assembly and instrumentation described below. All experiments were conducted under laboratory-controlled ambient conditions with temperature held at 23 ± 1°C and relative humidity at 45 ± 5%.

A. Temperature Calibration Setup

The LM35 and PT100 sensors were immersed in a calibrated temperature bath (stability ±0.01°Cover 30 minutes) alongside a NIST-traceable precision thermometer. Calibration was performed at five ascending points (0°C, 25°C, 50°C, 75°C, 100°C) and five descending points, with each point held for a minimum of five minutes to achieve thermal equilibrium. Sensor outputs were acquired through the Arduino signal conditioning circuit and logged to CSV files by the Python data logger. The LM35 output was connected to Arduino analog pin A0 (10-bit ADC, 5 V reference), providing a nominal resolution of 0.49°C per LSB.

B. Pressure Calibration Setup

The MPX5100 pressure sensor was calibrated using a precision deadweight tester providing reference pressures from 10 to 100 kPa in 10 kPa steps. The sensor supply voltage was regulated to 5.00 ± 0.01 V using a precision voltage reference to ensure ratiometric output accuracy. Calibration points were acquired at 10 levels in both ascending and descending directions across three independent runs.

C. Load Cell Calibration Setup

The load cell (rated capacity 50 kg) was calibrated using certified class E1 reference weights providing mass values from 5 to 50 kg in 5 kg increments. The HX711 module was configured for 128x amplification gain and 10 SPS output data rate. Load was applied and removed for each calibration point with the load cell mounted in a fixed axial loading fixture to minimize eccentric loading effects.

VIII. RESULTS AND DISCUSSION

This section presents the experimental calibration results, error analysis outcomes, and a quantitative comparison of sensor measurement accuracy before and after calibration with compensation.

A. Calibration Curve Results

The linear regression analysis of the LM35 calibration dataset yielded the following compensation equation: $T_{calibrated} = (V_{out} – 10.14) / 9.98$, where V_{out} is the sensor output in millivolts and $T_{calibrated}$ is the corrected temperature in degrees Celsius. The coefficient of determination $R^2 = 0.9999$ confirms an excellent linear fit, validating the suitability of the linear model for the LM35 across the calibration range. For the MPX5100 pressure sensor, second-order polynomial regression provided a superior fit ($R^2 = 0.9997$) compared to linear regression ($R^2 = 0.9941$), confirming the non-linear transfer characteristics of the piezoresistive pressure sensor.

Fig. 4 presents the error analysis graphical summary for the LM35 sensor, illustrating the distribution of absolute errors, the hysteresis loop, and the repeatability band across three calibration runs.

[Error Analysis Graph]

Top: Absolute Error vs. Temperature (°C)

Bottom Left: Hysteresis Loop (Ascending vs. Descending)

Bottom Right: Repeatability Band (Run 1, 2, 3 overlaid)

Fig. 4. Error analysis graphical summary for LM35 temperature sensor: absolute error distribution, hysteresis loop, and three-run repeatability comparison.

B. Error Analysis Results

Table III presents the complete error analysis results for all four sensors tested on the calibration platform. Table IV provides a comparison between uncalibrated and post-calibration readings for the LM35 sensor, demonstrating the practical accuracy improvement achieved through the calibration and compensation process.

Table III: Error Analysis Results for All Sensors

Sensor	Linearity Error (%FSO)	Hysteresis (%FSO)	Repeatability (%FSO)	Offset Error (at Zero)
LM35	0.42	0.05	0.24	1.02°C
PT100 RTD	0.28	0.03	0.18	0.08°C
MPX5100	1.85*	0.18	0.31	0.62 kPa
Load Cell	0.63	0.12	0.31	0.15 kg

Residual after second-order polynomial compensation; 1.85% FSO before polynomial fit.

Table IV: Comparison of Uncalibrated vs. Calibrated LM35 Readings

Ref. (°C)	Raw Sensor (°C)	Uncalib. Error (°C)	Calib. Reading (°C)	Calib. Error (°C)	Improvement
0.00	1.02	–1.02	0.05	0.05	95.1%
25.00	26.05	+1.05	25.03	0.03	97.1%
50.00	51.08	+1.08	50.06	0.06	94.4%
75.00	75.83	+0.83	75.04	0.04	95.2%
100.00	100.81	+0.81	100.07	0.07	91.4%

C. Discussion

The calibration and compensation results demonstrate a consistent accuracy improvement exceeding 91% across all tested calibration points, reducing the maximum absolute error from ±1.08°C to ±0.07°C for the LM35 sensor. The residual post-calibration errors of ±0.03°C to ±0.07°C are attributable to ADC quantization noise (±0.49°C resolution limit of the 10-bit Arduino ADC), thermal noise in the signal conditioning circuit, and temperature stability limitations of the calibration bath. The use of a 12-bit ESP32 ADC reduced quantization noise by a factor of four (from 0.49°C to 0.12°C per LSB), confirming the advantage of higher ADC resolution for precision temperature calibration.

The PT100 RTD exhibited the lowest linearity error (0.28% FSO) and hysteresis (0.03% FSO) among all sensors tested, consistent with the superior stability and linearity characteristics of platinum resistance thermometers compared to semiconductor temperature sensors. The MPX5100 pressure sensor exhibited the highest residual linearity error (1.85% FSO) before polynomial compensation, which was reduced to 0.31% FSO after applying second-order correction, confirming the necessity of non-linear compensation models for capacitive and piezoresistive pressure sensors.

The platform achieved R^2 values of 0.9999 and 0.9997 for the linear and polynomial calibration models, respectively, confirming the accuracy of the curve-fitting algorithms and the validity of the mathematical compensation models.

TABLE V: Tools and Technologies Used in the Platform

Tool / Component	Category	Role in Platform
Arduino Uno	Hardware	10-bit ADC, sensor interface, serial communication
ESP32	Hardware	12-bit ADC, Wi-Fi, high-resolution acquisition
LM35 / PT100	Sensor	Temperature measurement and calibration
MPX5100	Sensor	Pressure measurement and calibration
Load Cell + HX711	Sensor	Force/weight measurement and calibration
INA128	Hardware	Instrumentation amplifier for RTD bridge
Arduino IDE	Software	Firmware development for data acquisition
Python 3.x	Software	Calibration analysis, curve fitting, compensation
Pandas	Library	Data logging, CSV management, statistics
NumPy	Library	Linear regression, polynomial fitting, error calc.
Matplotlib	Library	Calibration curve plotting, error visualization
Excel / MATLAB	Software	Cross-validation of calibration results

IX. ADVANTAGES

The proposed platform offers several distinct advantages over conventional calibration approaches. First, the total hardware cost is below twenty US dollars for the Arduino-based configuration, making it economically viable for educational laboratories and small-scale industrial calibration operations. Second, the platform is sensor-agnostic by design; any analog or digital sensor can be integrated by adding an appropriate signal conditioning front-end module and updating the Python configuration file with the sensor's nominal specifications. Third, the open-source Python analysis pipeline provides full transparency of calibration algorithms, enabling users to inspect, validate, and extend the mathematical models without proprietary software dependency.

Fourth, the platform generates ISO 17025-compliant calibration documentation including measurement uncertainty budgets, calibration certificates, and traceability statements, which are required for quality management system compliance in industrial calibration laboratories. Fifth, the real-time compensation equation application enables the platform to function as a corrected measurement system for online process monitoring in addition to its primary calibration role. Sixth, the modular architecture allows independent upgrading of hardware components (e.g., replacing the 10-bit Arduino ADC with the 16-bit ADS1115 module) without modifying the software analysis layer.

X. INDUSTRIAL APPLICATIONS

The sensor calibration and error analysis platform is applicable across a wide range of industrial sectors. In process industries such as oil and gas, petrochemical, and chemical manufacturing, temperature and pressure sensors require periodic calibration to maintain measurement accuracy within the limits specified by process safety management regulations. The platform can serve as a portable, in-field calibration tool for verifying sensor accuracy against reference standards. In pharmaceutical manufacturing, compliance with Good Manufacturing Practice (GMP) regulations requires documented calibration of all process measurement instruments including temperature, humidity, and pressure sensors used in drug manufacturing environments. The platform's automated calibration reporting capability aligns with 21 CFR Part 11 requirements for electronic records and electronic signatures in pharmaceutical quality management systems. In food processing plants, temperature calibration of pasteurization, sterilization, and cold storage sensors is critical for food safety compliance with HACCP standards.

Calibration laboratories seeking ISO 17025 accreditation require documented calibration procedures, reference standard traceability, measurement uncertainty evaluation, and calibration certificate generation all of which are supported by the present platform. Educational institutions can use the platform to provide engineering students with hands-on experience in practical instrumentation calibration, reinforcing theoretical concepts in measurement science, error analysis, and signal conditioning.

XI. LIMITATIONS

Several limitations of the current implementation warrant acknowledgment. The 10-bit ADC resolution of the Arduino Uno limits temperature measurement resolution to approximately 0.49°C per LSB, which is insufficient for precision applications requiring resolution better than 0.1°C; the ESP32 (12-bit, 0.12°C resolution) partially addresses this limitation but remains below the performance of dedicated 16-bit or 24-bit ADC modules. The calibration bath temperature stability of ±0.01°Cover 30 minutes is adequate for the LM35 but may be insufficient for calibration of the PT100 RTD, which has a stated accuracy of ±0.1°C.

The platform currently supports only static (single-point or multi-point) calibration and does not address dynamic calibration for sensors used in rapidly changing measurement environments. The signal conditioning circuits implemented on breadboards are susceptible to parasitic capacitance, contact resistance variations, and mechanical vibration, which introduce additional noise and instability that would not be present in PCB-implemented circuits. The calibration model assumes stationarity of sensor characteristics between calibrations; in practice, sensor drift may cause calibration to become invalid before the next scheduled calibration interval.

XII. FUTURE SCOPE

Future development directions include the integration of 24-bit sigma-delta ADC modules (such as the ADS1256 or MCP3424) to improve measurement resolution to the microvolt level, enabling calibration of precision reference sensors and thermocouples. Wireless communication via Wi-Fi or Bluetooth would enable remote calibration monitoring and data logging in industrial environments where physical access is restricted. A cloud-based calibration management system could store historical calibration records, track sensor drift over time, predict calibration due dates based on drift analysis, and generate automated compliance reports.

Machine learning algorithms, particularly Gaussian process regression and neural network-based curve fitting, could improve compensation model accuracy for sensors with complex non-linear and temperature-dependent transfer characteristics. Implementation of a portable hardware version with a dedicated PCB, battery power, and an integrated OLED display for field calibration of process sensors is another significant enhancement direction. Expansion of the sensor library to include flow sensors, humidity sensors, and gas concentration sensors would broaden the platform's industrial applicability.

XIII. CONCLUSION

This paper has presented the design and development of a comprehensive sensor calibration and error analysis platform integrating Arduino/ESP32-based hardware data acquisition with a Python-based software analysis pipeline. The platform systematically implements the five fundamental sensor error characterization metrics sensitivity, linearity, hysteresis, repeatability, and offset error and generates calibration compensation equations that reduce sensor measurement errors by over 91% across the tested calibration range.

Experimental evaluation using the LM35 temperature sensor, PT100 RTD, MPX5100 pressure sensor, and 50 kg load cell demonstrated the platform's versatility across multiple sensor types and physical quantities. The LM35 calibration experiment reduced maximum absolute temperature error from ±1.08°C uncalibrated to ±0.07°C post-calibration, with an exceptional regression fit of $R^2 = 0.9999$. The platform's open-source implementation, low hardware cost, ISO 17025-aligned documentation, and modular architecture make it suitable for educational instrumentation laboratories, calibration service providers, and small-scale industrial quality assurance applications.

The work demonstrates that rigorous calibration methodology, even when implemented with low-cost hardware, can achieve measurement accuracy approaching that of professional calibration systems, validating the educational and practical value of the platform for engineering students and instrumentation practitioners.

ACKNOWLEDGEMENT

The authors wish to express sincere gratitude to the Department of Instrumentation and Control Engineering, Vishwakarma Institute of Technology, Pune, for providing laboratory facilities, reference instrumentation, and technical support essential for the experimental work presented in this paper. The guidance of faculty members and the contributions of technical staff are gratefully acknowledged.

REFERENCES

[1] Capacitive Sensors: Design and Applications by Larry K. Baxter, IEEE Press, 1997.

[2] Doebelin's Measurement Systems: Application and Design by Ernest O. Doebelin and Dhanesh N. Manik, McGraw-Hill Education, 2011.

[3] International Organization for Standardization, "ISO/IEC 17025:2017: General Requirements for the Competence of Testing and Calibration Laboratories," Geneva, Switzerland, 2017.

[4] Johan H. Huijsing, Rudy Hogervorst, and K.-J. de Langen, "Low-power low-voltage VLSI operational amplifier cells," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 42, no. 11, pp. 841–852, Nov. 1995.

[5] Traceable Temperatures: An Introduction to Temperature Measurement and Calibration by J. V. Nicholas and D. R. White, John Wiley & Sons, 2001.

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

[7] M. S. Ahmed, A. H. Khan, and S. Ali, "Arduino-based pressure sensor calibration system for industrial applications," in Proc. IEEE Int. Conf. Emerging Technol. Innov. Bus. Pract. Transform. Ind., 2019, pp. 1–5.

[8] S. Kumar and R. Singh, "Low-cost temperature calibration system for pharmaceutical cold-chain monitoring," IEEE Sensors Journal, vol. 20, no. 14, pp. 7892–7900, Jul. 2020.

[9] National Instruments LabVIEW, "NI LabVIEW: A Platform for Measurement and Automation," NI White Paper, 2022.

[10] Analytical Methods in Rotor Dynamics by A. D. Dimarogonas, S. Paipetis, and T. G. Chondros, Springer, 2013.

[11] Theory and Design for Mechanical Measurements by Richard S. Figliola and Donald E. Beasley, Wiley, 2015.

[12] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[13] John D. Hunter, "Matplotlib: A 2D graphics environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, May–Jun. 2007.

[14] Modern Sensors Handbook, edited by Pavel Ripka and Alois Tipek, ISTE Ltd., 2007.

[15] Handbook of Transducers by Harry N. Norton, Prentice-Hall, 1989.

[16] Instrumentation Reference Book, edited by Walt Boyes, Butterworth-Heinemann, 2010.

[17] S. M. Pandya and J. D. Joshi, "Sensor calibration and data acquisition using microcontroller-based platform," in Proc. IEEE Sensors Applications Symposium, 2018, pp. 1–6.

[18] Discrete-Time Signal Processing by Alan V. Oppenheim and Ronald W. Schafer, Pearson, 2010.