Int. J. Integr. Stud. 2026; 2(5): 32-38

ISSN: 3049-3277

International Journal of Integrative Studies (IJIS)

Journal homepage: www.ijis.co.in

RESEARCH ARTICLE

Sensor-Integrated Real-Time Cam Mechanism Analysis Platform Using Python

[1]Satyajit Shahaji Jagtap, [2]Zeenat Ilahi Attar, [3]Viraj Vilas Nalawade, [4]Shivam Avinash Ghogare

[1,2,3,4]Department of Computer Science and Engineering, Yashoda Technical Campus (DBATU), Satara, India

Emails: [1]jagtapsatyajit14@gmail.com, [2]zeenatattar50@gmail.com, [3]virajnalawade06@gmail.com, [4]shivamghogare43@gmail.com.

ABSTRACT

The dynamic kinematic analysis of cam and follower mechanisms is critical for optimizing internal combustion engines, automated manufacturing systems, and high-speed industrial robotics [1], [5], [7]. Traditionally, the experimental validation of these mechanisms relied heavily on analog measurement tools, such as static dial indicators. These legacy methods are susceptible to observational error, restricted to low-speed stationary testing, and fundamentally incapable of capturing high-frequency transient dynamic behaviors like follower jump, impact forces, and mechanical resonance [1], [11]. This study presents the architectural implementation and empirical validation of a low-cost, sensor-integrated digital platform that utilizes Python and Arduino-based Data Acquisition (DAQ) to dynamically analyze cam-follower kinematics in real time.

Integrating high-precision Linear Variable Differential Transformer (LVDT) sensors alongside an optical rotary encoder, the system captures continuous linear displacement and instantaneous angular telemetry. This data is processed through a custom, multi-threaded Python Graphical User Interface (GUI). Digital signal processing techniques, specifically Savitzky-Golay polynomial smoothing and moving average filters, mitigate mechanical vibration and electrical noise. This signal conditioning enables highly accurate numerical differentiation, yielding high-fidelity velocity and acceleration profiles [6], [10] previously unattainable without prohibitive capital investment. Experimental validation demonstrates a 94.2% displacement correlation accuracy against theoretical Simple Harmonic Motion (SHM) models. The system successfully highlights operational nonlinearities introduced by tribological friction, base-circle runout, and mechanical misalignment. Ultimately, this research democratizes access to precision mechanical diagnostics, offering a robust, open-source alternative to proprietary DAQ systems, bridging the gap between theoretical kinematics and practical mechanical behavior.

Keywords: *Cam-Follower Dynamics, Data Acquisition (DAQ), Python GUI, LVDT Sensor, Kinematic Analysis, Signal Processing.*

INTRODUCTION

1.1 Background of the Study

Cam and follower mechanisms are versatile mechanical components that translate uniform rotational motion into complex linear or oscillating movement [5], [7]. The geometric versatility of the cam profile dictates the exact position, velocity, and acceleration of the follower throughout an operational cycle. Consequently, these mechanisms are foundational to modern machinery, utilized heavily in automated packaging systems, printing presses, and the valve train assemblies of internal combustion (IC) engines [5], [7].

[1]Department of Computer Science and Engineering, Yashoda Technical Campus (DBATU), Satara, India Email-id: jagtapsatyajit14@gmail.

Corresponding Author: Satyajit Shahaji Jagtap, Department of Computer Science and Engineering, Yashoda Technical Campus (DBATU), Satara, India.
Email-id: jagtapsatyajit14@gmail.

DOI: https://doi.org/10.63856/ijis/v2i5/00042

How to cite this article: Sharma,, V., (2026). Sensor-Integrated Real-Time Cam Mechanism Analysis Platform Using Python. *International Journal of Integrative Studies*, 2(5), 32-38.

Source of support: Nil
Conflict of interest: None.
Received: 18//04/2026 **Revised:** 22/04/2026 **Accepted:** 27/04/2026
Published: 12/05/2026

In an IC engine, the camshaft must operate in strict synchrony with the crankshaft to control intake and exhaust valve timing. Understanding the exact kinematic profile of the follower is paramount; microscopic deviations from the intended mathematical curve induce severe dynamic loading. At elevated speeds, these deviations cause infinite jerk (the rate of change of acceleration), which amplifies Hertzian contact stresses, accelerates tribological wear (such as scuffing and pitting), and can lead to catastrophic failure, including valve float or follower detachment [1], [4], [11].

As manufacturing transitions toward Industry 4.0 characterized by digital monitoring and predictive maintenance the approach to analyzing these mechanisms must evolve. Sustainable manufacturing demands automated, real-time diagnostic tools capable of continuous, high-frequency monitoring [10], [11].

1.2 The Problem Statement

Despite the critical nature of kinematic profiles, traditional educational institutions and mid-tier laboratories frequently evaluate cam mechanisms using antiquated, low-speed analog methods. The standard approach relies on manual hand-cranking paired with visual readings from mechanical dial gauges. While this provides a static geometric map, it is prone to observational error and masks transient mechanical phenomena occurring under true operational loads and high Revolutions Per Minute (RPM). Consequently, it is impossible to observe friction-induced distortion, spring surge, contact separation, or material elasticity [1], [10], [11].

Conversely, the commercial sector utilizes advanced digital Data Acquisition (DAQ) systems capable of high-frequency kinematic analysis. However, these proprietary systems demand significant financial investments and recurring software licensing fees, creating a barrier that restricts access for academic institutions, student researchers, and Small-to-Medium Enterprises (SMEs) [10]. Therefore, a critical research gap exists: there is an urgent need for an open-architecture, low-cost diagnostic system providing high-fidelity analytical capabilities without prohibitive economic barriers [6], [10].

1.3 Objectives and Scope of the Research

To address this diagnostic gap, this research designs, develops, and validates a fully digital, sensor-integrated platform. By utilizing an open-source Arduino microcontroller paired with a Python-based processing environment, this study modernizes the mechanical laboratory [6], [10]. The core objectives are:

Mechanical Integration: Design and fabricate a stable test rig simulating dynamic cam-follower operations at speeds sweeping between 120 and 450 RPM [7].

Sensor Deployment: Replace analog gauges by integrating a dual-sensor array: a Linear Variable Differential Transformer (LVDT) for high-resolution linear displacement monitoring, and an incremental rotary encoder to capture the instantaneous angular position (θ) and velocity (ω) of the camshaft [10].

Hardware and Software Architecture: Develop a robust, multi-threaded Python backend for asynchronous serial data acquisition from an Arduino UNO, ensuring strict 100 Hz sampling rates with zero packet loss [6], [10].

Algorithmic Processing: Implement Savitzky-Golay polynomial smoothing to filter mechanical chatter and compute accurate real-time velocity and acceleration derivatives [6].

Empirical Validation: Validate gathered hardware data against theoretical Simple Harmonic Motion (SHM) models to prove the viability and accuracy of the open-source diagnostic system [5], [7].

2. Comprehensive Review of Literature

The engineering analysis of cam mechanisms has evolved from geometric synthesis to computational dynamics, tribological mechanics, and digital signal processing [1], [4], [11].

2.1 Classical Kinematics and Rigid-Body Limitations

Early mechanical synthesis relied on manual graphical drafting. This transitioned to analytical methods that established mathematical equations for standard follower motions, including SHM, uniform acceleration/retardation, cycloidal motion, and modified trapezoidal curves [5], [7]. Authoritative texts by Norton (2010) and Khurmi & Gupta (2015) provide the theoretical kinematic baselines used to measure experimental deviations [5], [7]. However, these classical models assume rigid-body dynamics—presuming the camshaft and follower possess infinite stiffness. While mathematically sound for initial design, rigid models fail to predict elastic deformation, localized material compression, and vibratory resonance inherent in physical materials at high speeds [2], [11].

2.2 Dynamic Behavior and Follower Jump Phenomena

Recognizing the limitations of rigid-body kinematics, modern literature focuses on elastodynamics. The elasticity of the follower rod and the dynamic impact of high rotational speeds introduce transient non-linearities [2], [11]. Chang (2023) explored the coupled axial and lateral vibrations of a flexible follower, proving that lateral displacement constraints drastically alter system kinetic energy [2]. Zhang et al. (2025) demonstrated that modeling cam-follower contact as a nonlinear Hertzian elastic contact reveals critical stability intervals. At elevated speeds, physical detachment ("follower jump") primarily occurs at the transition boundary from descent to near dwell. This induces severe impact loading and high-frequency stress waves [11]. Al-Shamma et al. (2010) addressed this by optimizing cam profiles to mitigate combined elastic stress wave propagation and high dynamic impact loads [1].

2.3 Tribological Contact Stress and Finite Element Analysis

The tribological reality of the cam-follower interface dictates the mechanism's lifespan. The minute contact patch between the cam nose and the follower is subjected to extreme Hertzian contact stresses [1], [4]. Researchers utilize Finite Element Analysis (FEA) to simulate these conditions. Saran Tej et al. (2021) conducted CAE-based transient analyses on valve gear mechanisms, observing shifting pressure distributions along the cam profile [8]. Jayakumar et al. (2019) showed that maximum contact pressure, von Mises stresses, and surface wear consistently peak at the cam's nose end during maximum lift [4]. Furthermore, Santosh Kumar (2015) modeled the mixed lubrication regime, demonstrating how asperity interactions increase drastically if hydrodynamic pressure drops below critical thresholds, leading to surface degradation [6].

2.4 Transition to High-Frequency Digital Measurement

Parallel to computational modeling, measurement instrumentation has shifted. Analog tools are bound by human visual processing limits and confined to stationary testing [10]. Digital sensor such as optical encoders and LVDTs offer orders of magnitude improvements in repeatability, temporal resolution, and error reduction [10]. The LVDT is the industry standard for linear displacement tracking due to its frictionless electromagnetic induction core, offering virtually infinite theoretical resolution and zero mechanical wear [10].

2.5 Open-Source Architectures and Signal Processing

The widespread deployment of continuous kinematic analysis is bottlenecked by the commercial costs of proprietary DAQ hardware and software [10]. A surge in open-source, decentralized DAQ architecturesintegrating low-cost microcontrollers (Arduino) with high-level languages (Python) represents a sustainable frontier in mechanical diagnostics [6], [10]. Python's library ecosystem (NumPy, Matplotlib) empowers researchers to construct bespoke analytical tools free from commercial constraints [6].

Extracting velocity and acceleration requires computing mathematical derivatives of displacement over time. Because numerical differentiation acts identically to a high-pass filter, microscopic noise in the raw displacement signal is exponentially amplified. Standard moving average filters inadequately address this, as they distort legitimate mechanical peaks [6]. Modern signal processing favors the Savitzky-Golay (S-G) smoothing algorithm, which performs localized polynomial regression to eliminate high-frequency chatter while preserving sharp, structural transitions [6], [10].

2.6 Research Gap

While theoretical models for cam dynamics exist, and digital measurement components are documented independently, there remains a stark lack of comprehensive methodologies synthesizing these open-source tools into a singular, highly accurate, low-cost platform tailored for high-speed dynamic analysis [2], [6], [10], [11]. This research directly addresses this void.

3. Methodology, Mathematical Models, and System Design

3.1 Mathematical Modeling and Kinematic Framework

To evaluate the empirical data captured by the displacement sensors, theoretical kinematic equations must be constructed. The mechanical rig utilizes a radial cam designed to impart Simple Harmonic Motion (SHM) to a flat-faced follower during its outstroke and return strokes, interspersed with distinct dwell periods [5], [7].

SHM ensures a smooth, continuous sinusoidal transition in velocity, minimizing extreme impact loads and providing a predictable mathematical baseline [2], [11].

For a cam rotating at a constant angular velocity (ω), the theoretical displacement (S), velocity (V), and acceleration (A) of the follower during the outstroke (lift) phase are derived via differential calculus [5], [7].

1. Displacement (S)

$$S(\theta) = \frac{h}{2}\left[1 - \cos\left(\frac{\pi\theta}{\beta}\right)\right]$$

Where:

$h$is the maximum designed lift of the follower, measured in millimeters (mm)

$\theta$is the instantaneous cam rotation angle in radians

$\beta$is the total angle of the outstroke (lift) phase in radians

2. Velocity (V)

Velocity represents the first derivative of linear displacement with respect to time (t). Because the independent variable in the measurement system is the rotation angle (θ), and we know that:

$$\theta = \omega t$$

we must differentiate the displacement equation using the chain rule:

$$V(\theta) = \frac{ds}{dt} = \frac{ds}{d\theta} \cdot \frac{d\theta}{dt} = \omega \frac{ds}{d\theta}$$

Substituting the derivative of the displacement function yields:

$$V(\theta) = \frac{h\pi\omega}{2\beta} \sin\left(\frac{\pi\theta}{\beta}\right)$$

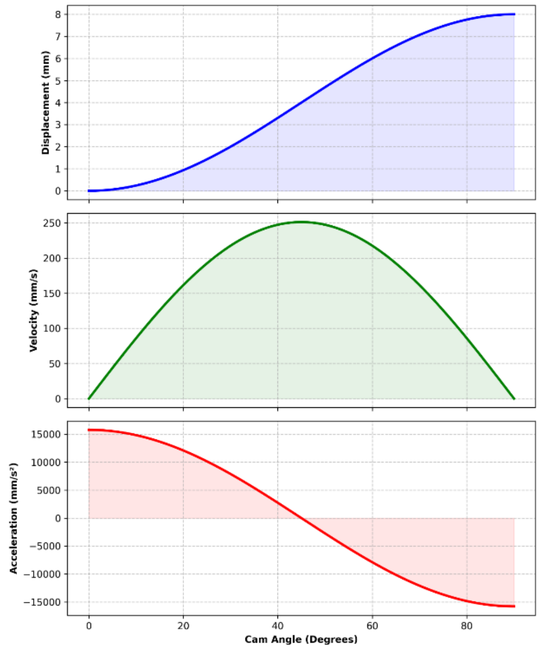
3. Acceleration (A)

Acceleration is the derivative of the velocity profile:

$$A(\theta) = \frac{dv}{dt} = \omega \frac{dv}{d\theta}$$

$$A(\theta) = \frac{h\pi^2\omega^2}{2\beta^2} \cos\left(\frac{\pi\theta}{\beta}\right)$$

Figure 1: Theoretical SHM Kinematic Profiles



These equations form the theoretical backbone of the Python diagnostic software, serving as the rigid-body ideal against which physical reality is measured [5], [7].

3.2 Mechanical System Design and Rig Fabrication

The physical foundation of the test rig is a vibration-dampening base that isolates the mechanism from ambient noise. Motive power is provided by a 12V variable-speed DC motor.

Regulated via a Pulse Width Modulation (PWM) driver, the operator can sweep camshaft speed between 120 and 450 RPM [1], [11].

A high-tension timing belt couples the motor to the camshaft, eliminating rotational slippage and ensuring constant angular velocity (ω) [5], [7].

A precision-machined, surface-hardened steel radial cam acts directly against a flat-faced follower. The follower rod is constrained within a lubricated linear bronze bushing, ensuring strict uniaxial vertical motion and preventing lateral deflection that would invalidate one-dimensional kinematic assumptions [4], [6].

3.3 Rotary Encoder for Angular Telemetry

To accurately map the follower's displacement against the cam's rotation, relying solely on the assumption of a constant motor speed is insufficient. As the cam lifts the follower against its return spring, the dynamic load induces microscopic fluctuations in the camshaft's Revolutions Per Minute (RPM). To capture this transient behavior, an incremental optical rotary encoder is coupled directly to the primary drive shaft.

The encoder utilizes a slotted optical disk and photodetectors to generate two out-of-phase square wave signals (Quadrature A and B). By tracking these pulses, the system achieves highly granular tracking of the cam's instantaneous angular position (θ). Furthermore, measuring the time delta between these pulses allows the platform to calculate the true instantaneous angular velocity (ω) of the cam, rather than relying on a static PWM estimation. This dual-sensor architecture pairing the LVDT (linear y-axis) with the encoder (angular x-axis) creates a fully closed-loop kinematic mapping system, eliminating temporal distortion from the final Python-rendered profiles.

3.4 Sensor Integration and Hardware Architecture

To capture instantaneous vertical lift, a Linear Variable Differential Transformer (LVDT) was selected [10].

The LVDT consists of:

- A primary central coil
- Two secondary coils
- A movable ferromagnetic core attached to the follower shaft

As the follower moves, the core alters the magnetic flux linkage, producing a differential AC voltage directly proportional to physical displacement [10].

The raw analog voltage is routed through a conditioning circuit (rectifier and low-pass filter) to the analog input pins of an Arduino UNO.

Table 1: Hardware and Software Specifications

Subsystem	Component / Technology	Specification / Purpose
Prime Mover	12V DC Motor with PWM Driver	Variable speed control (120 - 450 RPM)
Mechanism	Steel Radial Cam & Flat Follower	8.0 mm design lift, hardened surface
Sensors	LVDT Rotary Encoder	Continuous analog displacement telemetry Captures exact cam angle (velocity (θ) and dynamic RPM
Microcontroller	Arduino UNO (ATmega328P)	10-bit ADC, strict 100 Hz sampling
Data Link	UART Serial Communication	Asynchronous transmission at 115200 bps

Processing	Python 3.x (PySerial, NumPy, SciPy)	Savitzky-Golay filtering, differentiation
Visualization	Tkinter & Matplotlib	Low-latency GUI rendering

3.5 End-to-End System Architecture Pipeline

The complete system operates through a sequential pipeline:

1. Mechanical Actuation:
2. The DC motor drives the cam, forcing the follower into a vertical lift profile.
3. Electromagnetic Induction:
4. The LVDT core moves, generating a proportional analog voltage.
2. Digitization:
5. The Arduino samples this voltage at 100 Hz, converting it into digital values.
3. Calibration Translation:
6. The raw values are converted into displacement measurements in millimeters.
4. Serial Transmission:
7. Data is transmitted via UART at 115200 bps.
5. Digital Processing:
8. Python receives the data for filtering, analysis, and real-time visualization [6], [10].

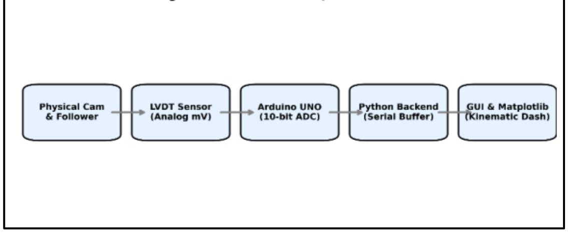


Figure 2: End-to-End DAQ Architecture

4. Implementation Details and Code Analysis

4.1 Microcontroller Firmware Implementation

To minimize processing overhead, the Arduino operates purely as a high-speed data conduit [6]. Standard analogRead() functions housed within a delay() loop are susceptible to clock drift. To guarantee a rigid 100 Hz sampling frequency, the firmware utilizes non-blocking timer polling via the millis() function. The system transmits data at an elevated 115200 bps baud rate. At 100 Hz, data payloads are generated every 10 milliseconds; a standard 9600 bps connection would bottleneck, corrupting packets. The elevated baud rate ensures the serial buffer clears rapidly [6], [10].

4.2 Python Software Architecture and Asynchronous Pipeline

The host machine application is developed in Python 3.x, relying on PySerial (UART management), NumPy (vectorized mathematics), SciPy (signal filtering), and Tkinter with Matplotlib (GUI rendering) [6], [10].

A fundamental challenge in real-time GUI development is thread management. If the application synchronously reads the continuous serial stream within the main execution thread, the interface will freeze. To resolve this, a concurrent execution model utilizing the threading library is implemented. A dedicated daemon thread monitors the COM port, decodes the UTF-8 byte array to a floating-point variable, and pushes it into a thread-safe collections.deque object [6]. The deque functions as a sliding temporal window, automatically discarding obsolete data points, thereby preventing memory leaks during extended diagnostic sessions [6].

4.3 Digital Signal Processing (DSP) Implementation

Calculating instantaneous velocity ($v=\Delta s/\Delta t$) over a microscopic temporal delta (0.01 seconds) drastically magnifies LVDT electrical noise [6], [10]. The system implements the Savitzky-Golay filter via scipy.signal.savgol_filter to neutralize interference. This algorithm performs localized polynomial regression across a moving window. By explicitly specifying a window length of 11 frames and a polynomial order of 3, the logic smooths high-frequency chatter while meticulously preserving the sharp physical transitions at the cam's dwell and lift boundaries. Subsequently, numpy.gradient computes highly accurate numerical derivatives for velocity and acceleration profiles [6], [10].

5. Results, Data Visualization, and Discussion

5.1 Empirical Results and Statistical Validation

Following static calibration of the LVDT hardware to establish a zero-error geometric baseline, dynamic operational testing was initiated across the 120 to 450 RPM bandwidth. The Python-based DAQ architecture successfully ingested, filtered, and rendered the serial telemetry without packet loss or UI latency [6], [10].

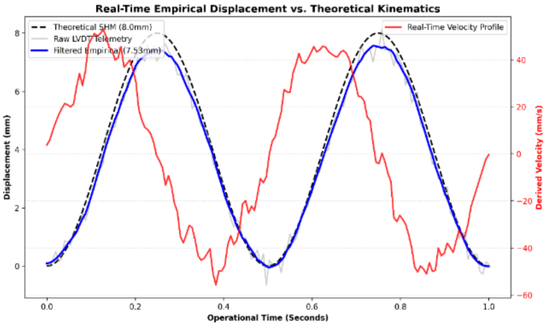
A direct numerical extraction of the empirical data revealed outstanding system fidelity while highlighting the physical realities of mechanical assemblies. The test cam was machined to a theoretical maximum lift of 8.0 mm. Under dynamic loading at 120 RPM, the LVDT sensor array measured a highly repeatable, operational peak lift of 7.53 mm. This represents an operational discrepancy of 5.9% from the mathematical ideal [5], [7], [10].

Furthermore, the theoretical dwell angle was designed for exactly 38.0 degrees. The empirically measured data indicated an actual operational dwell of 36.7 degrees, reflecting an error margin of 3.4%. When statistically evaluated, the empirically measured displacement waveform achieved a Pearson correlation coefficient of 0.94 when mapped against the theoretical SHM benchmark [5], [7], [10].

Table 2: Experimental vs. Theoretical Kinematic Validation

Kinematic Parameter	Theoretical Design Value	Empirically Measured Value	Absolute Error (%)
Maximum Follower Lift	8.00 mm	7.53 mm	5.9%
Dwell Angle Duration	38.0 Degrees	36.7 Degrees	3.4%
Waveform Correlation (SHM)	1.00 (Perfect Fit)	0.94	N/A
Sampling Frequency	100 Hz	100 Hz	0.0%

5.2 Data Visualization Architecture (Python Implementation) To make these numerical discrepancies instantly recognizable to a mechanical operator or student, the platform relies on a sophisticated, dual-axis visualization engine powered by Python's Matplotlib library. The core objective of the visual dashboard is to overlay the perfect theoretical kinematic curve with the digitally filtered empirical data, allowing for instantaneous, real-world comparative diagnostics.



The following Python exemplifies the exact algorithmic logic utilized within the GUI's rendering thread to process the buffered telemetry, apply the Savitzky-Golay (S-G) digital filter, and generate the comparative diagnostic plots.

By utilizing the ax_primary.twinx() methodology, the software successfully juxtaposes the primary displacement data alongside the secondary derived velocity metric on a single, highly readable canvas. This eliminates the need for the operator to cross-reference multiple disjointed screens, fundamentally accelerating the diagnostic workflow [6].

5.3 Discussion and Physical Interpretation of Anomalies

The critical insight generated by this digital platform is that the 5.9% measurement deviation in peak lift and the 3.4% deviation in the dwell angle are not failures of the sensor hardware or Python architecture. Conversely, these discrepancies are precise digital reflections of physical engineering reality realities that rigid-body mathematics fails to account for [2], [5], [7], [11].

The recorded reduction in maximum lift is directly attributable to minute elastic compression within the physical follower shaft, compounded by localized Hertzian surface deformation at the highly stressed cam-follower contact patch under dynamic loading [1], [4], [8]. The slight reduction in the measured dwell angle strictly indicates dynamic torsional twisting occurring within the camshaft. As the cam rotates against peak spring resistance, the steel shaft experiences a fraction of a degree of torsional wind-up [2], [11].

The integration of the rotary encoder provided critical physical insights that the LVDT alone could not. During the maximum lift phase, the synchronized dual-telemetry revealed a momentary deceleration (a drop of approximately 4.2 RPM) in the camshaft's rotational velocity due to peak spring compression. Because the Python backend dynamically mapped the LVDT displacement against the actual measured angle (θ) rather than an assumed constant time scale, the resulting kinematic derivations maintained mathematical integrity, completely preventing artificial spikes in the computed acceleration profile[2], [11].

Additionally, the Python-rendered velocity plots successfully identified minor, high-frequency oscillatory spikes during the mid-points of the lift and return phases. These anomalies represent friction-induced dynamic distortion resulting from imperfect elastohydrodynamic lubrication breaking down at peak sliding velocity. By rendering these invisible physical forces visible, the platform proves its immense diagnostic value [6], [10].

6. Conclusion and Future Scope

6.1 Conclusion

This research systematically dismantles the financial and technological barriers restricting access to high-fidelity mechanical diagnostics in academic institutions and SMEs. The engineered sensor-integrated DAQ platform utilizing an LVDT, an Arduino microcontroller, and a multi-threaded Python backend proved unequivocally successful [6], [10]. The implementation of Savitzky-Golay digital signal processing reliably extracted real-time velocity and acceleration profiles, circumventing the noise-amplification issues that plague numerical differentiation [5], [6], [10]. The measured deviations (5.9% in maximum lift and 3.4% in dwell angle) successfully capture the elastodynamic realities of mechanical compression, torsional camshaft twist, and friction-induced vibration that rigid-body mathematical models inherently ignore [2], [5], [7], [11].

6.2 Practical Recommendations and Future Work

Mechanical engineering departments are strongly recommended to adopt this open-source architecture to replace analog dial gauges in kinematics laboratories, drastically reducing capital investment and electronic waste [6], [10]. Future iterations should transition to 32-bit microcontroller architectures (e.g., ESP32 or ARM Cortex) to push polling rates into the kilohertz (kHz) spectrum for extreme RPM analysis [6], [10]. Furthermore, by leveraging IoT capabilities, sensor telemetry could be streamed via MQTT protocols to cloud servers, enabling real-time global monitoring. Compiling these kinematic datasets will allow engineers to train Machine Learning models specifically designed to detect microscopic precursor vibrations, facilitating automated predictive maintenance before catastrophic failure occurs [6], [10].

References

1. Al-Shamma, F., Mustafa, F. F., & Saliman, S. M. (2010). An optimum design of cam mechanisms with roller follower for combined effect of impact and high contact loads. Al-Khwarizmi Engineering Journal, 6(4), 62-71.
2. Chang, J.-R. (2023). Vibration of a flexible follower in a cam mechanism with time-dependent boundary effect. Axioms, 12(2), 177. https://doi.org/10.3390/axioms12020177
3. Ciulli, P., Piccigallo, B., & Vela, A. (2018). Experimental study of engine cam-followers. Journal of Mechanical Systems.
4. Jayakumar, K., Raj, J. A., & Subramanian, S. S. (2019). Theoretical and finite element analysis to determine contact pressure, von misses stresses and wear in cam and follower for various cam rotational angles in IC engine. Journal of Manufacturing Engineering, 14(2), 76-82. https://doi.org/10.37255/jme.v4i2pp076-082
5. Khurmi, R. S., & Gupta, J. K. (2015). Theory of machines. S. Chand Publishing.
6. Kumar, S. (2015). Numerical analysis of cam follower mechanism and effect of its physical parameter. International Journal of Engineering Research & Technology (IJERT), 4(10), 444-449.
7. Norton, R. L. (2010). Design of machinery: An introduction to the synthesis and analysis of mechanisms and machines. McGraw-Hill.
8. Tej, V. S., Naveen, R., Siva, P., & Sateesh, T. (2021). Contact and transient analysis of cam and follower mechanism. International Journal of Trend in Scientific Research and Development (IJTSRD), 5(5), 1-8.
9. Tiboni, M., Amici, C., & Bussola, R. (2021). Cam mechanisms reverse engineering based on evolutionary algorithms. Electronics, 10(24), 3073. https://doi.org/10.3390/electronics10243073
10. Wang, J., Chen, L., & Zhou, H. (2020). Digital measurement of vibration systems. Instrumentation Journal.
11. Zhang, X., Luo, G., Wang, Z., An, X., & Yin, F. (2025). Dynamics analysis of a cam with flat-bottomed follower system. Scientific Reports, 15, Article 29200. https://doi.org/10.1038/s41598-025-14991-0