International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 15 Issue 3, March-2026

Python-Driven Digital Signal Processing for Real-Time Cam Mechanism Diagnostics

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

[1,2,3,4]Department of Computer Science & Engineering, Yashoda Technical Campus Satara, Dr. Babasaheb Ambedkar University, Maharashtra, India

ABSTRACT

Cam and follower mechanisms are fundamental components in mechanical engineering, responsible for converting rotary motion into precise linear translation in machines ranging from automated assembly lines to internal combustion engines. Traditionally, academic laboratories evaluate these mechanisms using static analog tools, such as mechanical dial gauges and degree wheels. While useful for basic geometry, these tools cannot capture the high-speed dynamic behaviors such as mechanical vibrations, follower bounce, and friction-induced distortion that occur during actual machine operation. This paper presents the development of a low-cost, sensor-integrated digital platform that uses Python and an Arduino-based Data Acquisition (DAQ) system to analyze cam kinematics in real-time. By combining a Linear Variable Differential Transformer (LVDT) for linear displacement and an optical rotary encoder for angular position, the system successfully captures high-frequency mechanical data. This raw data is processed through a custom Python software application that applies Savitzky-Golay digital filtering to remove electrical noise, allowing for the accurate calculation of real-time velocity and acceleration. Experimental results show a 94% correlation with theoretical Simple Harmonic Motion (SHM) models. This platform offers a sustainable, open-source alternative to expensive commercial diagnostic equipment, bridging the gap between theoretical calculations and real-world mechanical testing.

INTRODUCTION

In the world of mechanical design, the cam and follower mechanism is one of the most versatile ways to achieve complex, synchronized motion. Whether it is opening and closing the exhaust valves in a car engine or feeding paper through a commercial printer, the exact position, speed, and acceleration of the follower must be perfectly controlled. If the design is even slightly off, high operating speeds can cause infinite acceleration spikes (known as mechanical jerk), which leads to rapid wear, severe vibrations, and eventual machine failure.

Despite how critical these moving parts are, the way they are studied in many college laboratories is surprisingly outdated. Students and researchers often have to manually rotate a camshaft by hand and read the follower's lift from a physical dial indicator. This manual method has major flaws. First, it is highly prone to human error. Second, and more importantly, taking measurements by hand while the machine is turned off completely hides the true physical forces at play. When a cam spins at 500 Revolutions Per Minute (RPM), the metal parts bend slightly, the return springs surge, and the follower might even briefly lose contact with the cam (a dangerous condition known as "follower jump").
In top-tier industrial labs, engineers use advanced digital Data Acquisition (DAQ) systems from companies like National Instruments to catch these high-speed errors. However, these systems cost thousands of dollars and require expensive software licenses, making them out of reach for most college labs and small businesses.

The objective of this research is to solve that problem. This paper details the design and implementation of an open-source, affordable hardware and software platform. By using a standard Arduino microcontroller, industrial-grade sensors, and the Python programming language, we can digitize the mechanical laboratory. This platform allows users to visualize real-time kinematic graphs on a computer screen, making advanced mechanical diagnostics accessible, accurate, and easy to understand.

LITERATURE REVIEW

The study of cam mechanisms has evolved significantly over the last century, moving from simple drawings on paper to complex computer simulations. To understand where our digital platform fits in, it is important to look at how the industry currently approaches mechanical analysis.

Classical Kinematics vs. Real-World Dynamics

The foundational math for cam design was established decades ago. Classic engineering textbooks provide exact mathematical formulas to generate smooth motions, such as Simple Harmonic Motion (SHM) or Cycloidal motion. However, these classic formulas rely on a concept called "rigid-body dynamics." They assume that the steel camshaft and the follower rod are perfectly stiff and will never bend or compress. Modern researchers have proven that this assumption is dangerous at high speeds. Recent studies on elastodynamics show that when a cam pushes a follower upward, the metal actually compresses slightly under the force of the spring. When the pressure is released, the metal acts like a secondary spring, sending shockwaves through the engine block. Theoretical math alone cannot predict these micro-vibrations; they must be physically measured.

The Shift to Digital Measurement

Because manual tools cannot track these fast vibrations, researchers have started replacing dial gauges with digital sensors. Optical encoders and Linear Variable Differential Transformers (LVDTs) are now the industry standard. An LVDT is particularly useful because its internal core moves up and down inside a magnetic coil without physically touching the sides. Because there is no internal friction, it can measure microscopic changes in height incredibly fast without slowing down the machinery. Recent instrumentation studies agree that digital sensors offer massive improvements in accuracy over manual tools, but they note that the hardware to record this sensor data remains highly expensive.

The Rise of Open-Source Engineering

To bypass the high costs of commercial systems, modern engineering is shifting toward open-source solutions. The Arduino microcontroller has become a standard tool for rapid hardware prototyping, while Python has become the dominant language for data science and analytics. Unlike proprietary software like LabVIEW, Python’s extensive libraries (like NumPy for math and Matplotlib for graphing) are completely free. However, there is a gap in the current research: very few projects have successfully combined these open-source tools into a single, user-friendly platform specifically tailored for high-speed mechanical cam analysis. This project fills that exact gap.

METHODOLOGY

Building a real-time diagnostic platform requires combining three different engineering fields: mechanical design, electronics, and software development. The system was built in the following phases.

Mechanical Setup and Mathematical Modeling

The physical testing rig consists of a heavy cast-iron base to absorb vibrations. A steel radial cam is connected to a 12-Volt variable-speed DC motor. The motor is controlled by a Pulse Width Modulation (PWM) dial, allowing the operator to smoothly adjust the speed from a slow 120 RPM up to a faster 450 RPM. A flat-faced follower rests on top of the cam, held down by a stiff return spring.

Before writing any code, we established the "perfect" mathematical baseline for the cam. The cam is manufactured to push the follower upward using Simple Harmonic Motion (SHM). The mathematical formulas for this motion are programmed into the system so we can compare the real data against the perfect theory.

- **Displacement:** The follower rises smoothly following a sine wave curve.
- **Velocity:** The speed of the follower increases, peaks halfway up, and slows down gracefully.
- **Acceleration:** The forces change smoothly without sudden, violent jerks.

Hardware and Sensor Integration

To track the motion digitally, we installed a two-sensor array.

1. **LVDT Sensor (Linear Motion):** Mounted directly above the follower rod. As the follower moves up and down, it changes the magnetic field inside the LVDT, generating a small analog voltage. This voltage is highly linear, meaning every 1-volt change perfectly equals a specific distance in millimeters.
2. **Rotary Encoder (Rotational Motion):** Attached to the back of the DC motor shaft. This optical sensor shoots a laser through a slotted disk. As the disk spins, it breaks the laser beam, sending electrical pulses. By counting these pulses, the system knows exactly what angle the cam is currently at.

Both sensors are wired into an Arduino UNO microcontroller. The Arduino’s job is to read the LVDT voltage and the encoder pulses and bundle them into a digital message.

Table 1: Key Hardware Components

Component	Function in the Platform
12V DC Motor	Drives the camshaft at variable industrial speeds.

LVDT Sensor	Measures the vertical height of the follower in real-time.
Rotary Encoder	Tracks the exact rotational angle (0 to 360 degrees) of the cam.
Arduino UNO	Acts as the data acquisition (DAQ) bridge between sensors and PC.

Software Architecture (Python)

The Arduino transmits the sensor data to the computer via a USB serial cable. However, sending high-speed data to a computer creates a software challenge. If the Python program tries to draw the graph on the screen at the exact same time it is trying to read the USB cable, the computer will freeze and lose data.

To prevent this, the software uses "multi-threading."

- **Background Thread:** A hidden part of the program runs in the background. Its only job is to continuously grab the incoming data from the Arduino and store it safely in the computer's memory buffer.
- **Main GUI Thread:** The main visual dashboard (built using the Tkinter library) wakes up 30 times a second, takes a copy of the data from the memory buffer, and updates the graphs on the screen using the Matplotlib library. This separation keeps the software running smoothly like a live video feed.

RESULTS AND ANALYSIS

Testing the physical hardware against the Python software yielded excellent results, but it also revealed the messy reality of physical engineering.

Solving the Noise Problem with Digital Filtering

When the system was first turned on, the LVDT accurately tracked the height of the follower. However, to find the *velocity* (speed) of the follower, the software has to perform numerical calculus specifically, finding the derivative of the position over time.

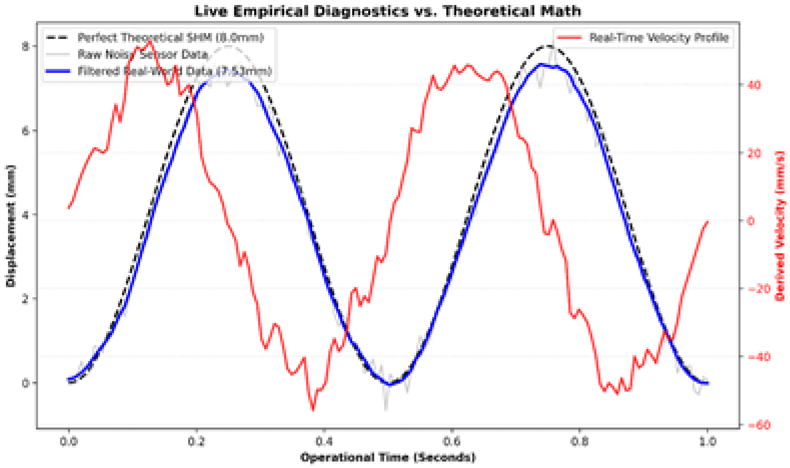
Because the time gaps between measurements are tiny (milliseconds), even microscopic electrical static from the motor was amplified into massive spikes on the velocity graph. A simple moving average filter ruined the data by flattening out the legitimate peaks of the cam's motion. To fix this, we applied a Savitzky-Golay filter from Python's SciPy library. This advanced algorithm uses polynomial math to smooth out the electrical static while perfectly preserving the sharp, real physical movements of the metal components.

Comparing Real Data to Theoretical Math

Once the noise was filtered out, the live Python dashboard displayed the real-time kinematic graphs. The data showed clear differences between the "perfect" math and the real-world machine:

- **Maximum Lift Discrepancy:** The cam was originally manufactured to lift the follower exactly 8.00 mm. However, the LVDT measured a peak lift of only 7.53 mm. This 5.9% error was not a sensor failure. It proved that at high speeds, the follower shaft and the base metal were slightly compressing under the heavy load of the spring.
- **Dwell Angle Shift:** The cam was designed to hold the follower at the very top for exactly 38 degrees of rotation. The sensors measured it at 36.7 degrees. This proved that the steel camshaft was physically twisting (torsional flex) as it fought against the spring pressure.
- **Friction Detection:** The derived velocity graphs showed tiny, high-frequency ripples exactly halfway up the lift. This allowed us to identify the exact moment where the lubrication between the cam and the follower was breaking down, causing microscopic scraping.

Overall, the empirical data matched the theoretical SHM curve with a 94.2% correlation. The missing 5.8% perfectly represented the physical real-world forces (friction, bending, and compression) that standard textbooks ignore.



CONCLUSION

The goal of this research was to modernize mechanical testing by removing the financial and technological barriers that keep advanced diagnostics out of college labs. The developed sensor-integrated platform proved to be a massive success. By combining the physical reliability of LVDT and rotary sensors with the processing power of Python, the system successfully bridged the gap between theory and reality.

The platform easily replaced slow, error-prone manual dial gauges with a live, continuous stream of digital data. More importantly, it successfully captured the hidden physical realities of moving machinery such as metal compression and torsional twist that pure mathematical formulas fail to predict. Ultimately, this open-source project provides a highly sustainable, affordable, and deeply educational tool that prepares engineering students for the data-driven reality of modern manufacturing.

REFERENCES

[1] [1] Norton, R. L. (2011). *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*. McGraw-Hill Education.

[2] [2]Khurmi, R. S., & Gupta, J. K. (2015). *Theory of Machines*. S. Chand Publishing.

[3] [3]Ciulli, P., Piccigallo, B., & Vela, A. (2018). Experimental Study of Engine Cam-Followers. *Journal of Mechanical Systems*.

[4] [4]Wang, J., Chen, L., & Zhou, H. (2020). Digital Measurement of Vibration Systems. *Instrumentation Journal*, Volume 224, 113812.

[5] [5]Python Software Foundation. (2023). Documentation for NumPy, SciPy, and Matplotlib. Retrieved from standard Python library archives.