

Remaining tool life monitoring system based on error and machine learning

Kush Joshi¹, Aryan Yardi², Anshika Goyal³, S Ahmed Hamdan⁴, and Saravanakumar R⁵

¹School of Electrical Engineering, Vellore Institute of Technology, Vellore, India

^{2,3,4,5} Department of Control and Automation, School of Electrical Engineering, Vellore Institute of Technology, Vellore, India

Abstract- Unplanned tool failure in CNC machining leads to poor surface quality, scrapped parts and machine downtime. This paper addresses the design and implementation of a low-cost, real-time system for monitoring tool wear and estimating the remaining useful life (RUL) of cutting tools. The hardware architecture comprises an ADXL345 3-axis accelerometer mounted on the spindle housing, an electret microphone for acoustic emission sensing and an ACS712 Hall-effect sensor for spindle motor current. The signals are conditioned by a custom low-pass filter board and digitised by an Arduino Mega or an NI myDAQ. The software is developed in LabVIEW and Python and applies DC offset removal, Daubechies db4 wavelet denoising, windowed FFT and statistical feature extraction (RMS, kurtosis, crest factor, tooth-passing amplitude and sideband energy). Tool condition is classified as Healthy, Early-Wear or Critical using calibrated thresholds, and RUL is estimated by extrapolating the polynomial trend of the RMS feature, with machine learning models identified as the next stage of development. A real-time dashboard with alert thresholds presents the results to the operator. The system is being validated on a benchtop CNC router against four known tool conditions, with targets of at least 90 % classification accuracy, a false alarm rate below 5 % and an RUL error below 10 %. By enabling the early detection of tool wear, this framework is intended to support predictive maintenance, reduce unplanned downtime and extend tool life in automated manufacturing.

Keywords: Tool wear monitoring, Remaining useful life (RUL) prediction, Predictive maintenance, CNC machining, Vibration analysis, Acoustic emission, Spindle current monitoring, Signal processing, Wavelet denoising, Fast Fourier transform (FFT), Feature extraction, Tool condition monitoring, Real-time monitoring, LabVIEW, Polynomial trend extrapolation.

I. INTRODUCTION

Cutting tools in CNC machining wear progressively as material is removed. A worn tool increases cutting forces, degrades surface finish and dimensional accuracy and, in the worst case, breaks during a cut and damages the workpiece and the spindle. Conventional practice either replaces tools at fixed intervals, which discards usable tool life, or runs tools until they fail, which risks costly stoppages. Both approaches ignore the actual condition of the tool. Consequently, there is an increasing demand for online and non-invasive methods that estimate tool condition during cutting [1, 2].

Tool wear leaves recognisable signatures in several measurable signals. Vibration amplitude rises as the cutting edge degrades [3], acoustic emission

responds to chipping and cracking [2], and the spindle motor draws more current as the cutting load increases [4]. Nonetheless, machining signals are noisy and non-stationary, and the raw amplitude of any single sensor is an unreliable indicator of wear. Reliable monitoring therefore requires several sensing modalities, effective noise removal and features that capture how the signal changes over time. In light of these requirements, this paper presents a virtual-instrumentation-based tool life monitoring system built with LabVIEW and Python. The system acquires vibration, acoustic emission and spindle current signals, applies wavelet denoising [5] and windowed FFT to isolate fault-related frequency components, and uses trend analysis to predict the remaining useful life of the tool. LabVIEW [6] provides the acquisition environment and the operator front panel. Public run-to-failure datasets such as the PHM 2010

challenge data [7] are identified for training machine learning models in future work.

II. THEORETICAL BACKGROUND

1.CNC machines use G-code

to drive multi-axis motion. During milling, each tooth entering the cut produces a periodic force, so vibration appears at the spindle frequency f_s and its harmonics, and at the tooth-passing frequency $z \cdot f_s$ for a cutter with z teeth.

Forced vibration from imbalance, misalignment and bearing defects adds further components, while chatter, a self-excited vibration caused by the regenerative effect, produces surface defects. Tool wear typically progresses through three stages: run-in, steady-state and rapid wear. Vibration amplitude increases as wear advances, and sidebands around the tooth-passing frequency indicate tool damage.

In the time domain, vibration signals are characterised by RMS, peak value, crest factor and kurtosis. In the frequency domain they are characterised by the spindle frequency, the tooth-passing frequency and the sideband energy. To avoid aliasing, the sampling frequency must satisfy $f_{\text{sample}} \geq 2 \times f_{\text{max}}$ (Nyquist criterion), and an anti-aliasing filter is applied before analog-to-digital conversion. Guidance on evaluating machine vibration severity is given in ISO 10816-3 [8].

2.Maintenance strategies and predictive maintenance

A maintenance strategy can be selected from the cost of failure and the ease of monitoring (Fig. 1). Run-to-failure suits assets with a low cost of failure that are hard to monitor, and time-based preventive maintenance suits costly failures that are hard to monitor. Condition-based maintenance suits easily monitored assets with a low cost of failure, while predictive maintenance (PdM) occupies the quadrant of high ease of monitoring and high cost of failure. CNC spindle health is therefore an ideal PdM candidate. PdM is generally reported to reduce unplanned downtime by up to 50 % compared with reactive maintenance, to extend tool life by 20–40 % through optimal

replacement timing and to lower maintenance cost by about 12 % compared with time-based preventive schedules. It also enables Industry 4.0 integration through the Industrial Internet of Things (IIoT) and edge analytics.

PdM techniques can be divided into statistical-based methods (regression analysis, time series analysis and reliability modelling) and condition-based methods (oil control, vibration monitoring, current analysis, thermal imaging and ultrasound monitoring), as shown in Fig. 2. The present work combines vibration, acoustic and current monitoring with regression and time-series trend analysis.

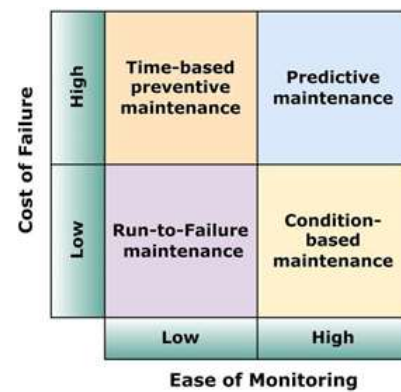


Fig. 1. Maintenance strategy selection matrix based on cost of failure and ease of monitoring

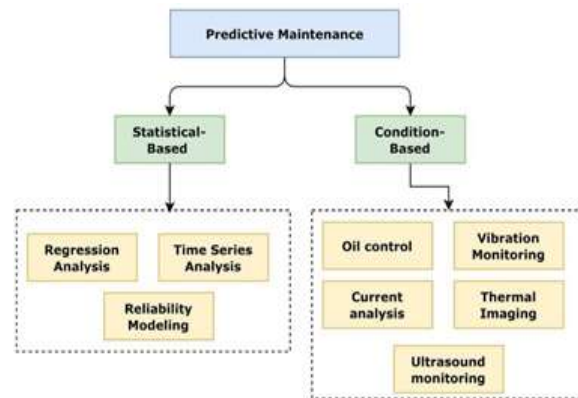


Fig. 2. Taxonomy of predictive maintenance techniques

3. Signal processing techniques

The following methods are used in the processing chain:

- Wavelet transform: multi-resolution time-frequency analysis that decomposes the signal into approximation (low-frequency) and detail (high-frequency) components. The Daubechies db4 wavelet is well suited to non-stationary machining signals.
- Windowed FFT (STFT): tracks the frequency content over time. A Hanning window minimises spectral leakage, and an FFT size of 1024–4096 points gives sufficient resolution.
- Statistical feature extraction: RMS, kurtosis, skewness and entropy.

Trend analysis: linear regression and exponential smoothing applied to the extracted features.

Threshold comparison: statistical process control (SPC) limits.

III. SYSTEM ARCHITECTURE AND HARDWARE DESIGN

The primary goal of the hardware architecture is to acquire clean, time-aligned signals from three sensing modalities so that the system can reach its targets of a detection rate above 90 % and a false alarm rate below 5 %. Fig. 3 shows the overall architecture, from the sensors on the CNC spindle to the host PC.

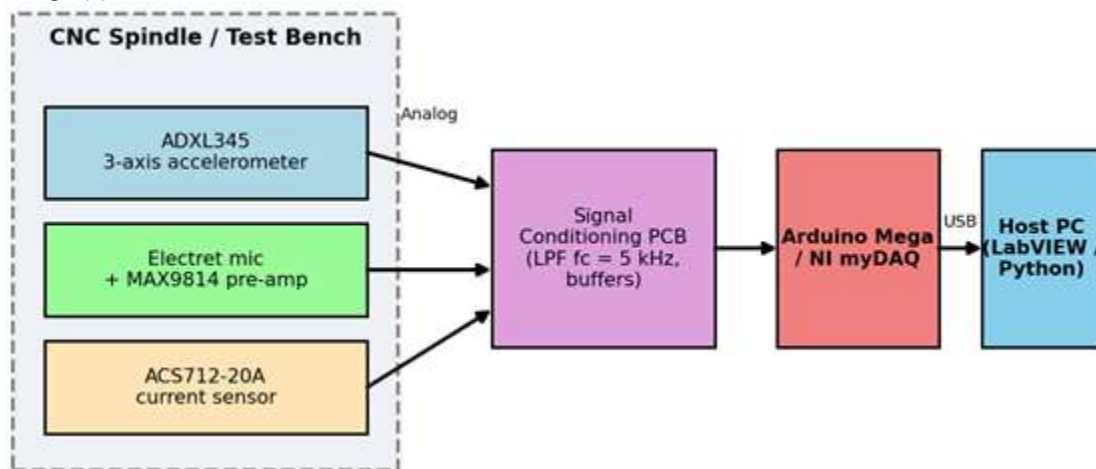


Fig. 3. Block diagram of the hardware architecture and sensor layer.

Materials

1. Sensor selection and mounting

- **ADXL345:** a digital 3-axis accelerometer (± 16 g) [9], stud-mounted on the spindle housing and connected to the Arduino Mega through the I²C/SPI interface.
- **Electret microphone:** acquires the acoustic emission signal to detect chipping or cracking. A MAX9814 auto-gain pre-amplifier feeds an ADC channel.
- **ACS712-20A:** a Hall-effect current sensor [10] placed inline with the spindle drive to correlate load variations with tool wear.

2. Signal conditioning and data acquisition

A custom signal-conditioning PCB designed in KiCad provides a low-pass filter ($f_c = 5$ kHz) and buffer amplifiers, which act as the anti-aliasing

stage before conversion. The Arduino Mega [11] digitises the channels with its 10-bit ADC at 10 kSa/s and streams the data to the PC over a USB-serial bridge. An NI myDAQ (± 10 V analog input, 200 kSa/s, 16-bit) is used as an optional higher-resolution unit for bench validation. The host PC runs LabVIEW and Python.

3. Calibration and enclosure

- **Accelerometer:** two-point calibration using gravity (0 g and 1 g).
- **Microphone:** sensitivity of -44 dBV/Pa, mapped to the dB SPL scale.
- **Current sensor:** zero-current offset trim and a sensitivity of 100 mV/A.

- **Enclosure:** the signal-conditioning PCB is housed in a DIN-rail enclosure rated IP54 for the industrial environment.

IV. SOFTWARE ARCHITECTURE

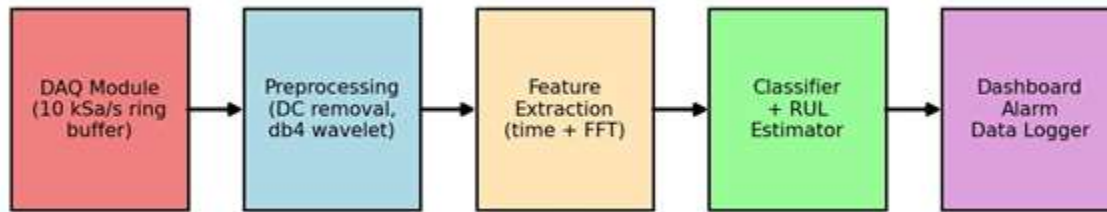


Fig. 4. Software processing pipeline of the tool life monitoring system

1. Data acquisition and preprocessing

The acquisition module reads the serial stream at 10 000 Sa/s into a NumPy ring buffer [12] holding 10 000 samples per channel, with timestamps written to CSV. The five channels (X-, Y- and Z-axis acceleration, microphone and current) are acquired in parallel, and the live plot is refreshed every 100 ms. Preprocessing removes the DC offset by subtracting the signal mean, applies wavelet denoising (db4, level 5, soft thresholding) and segments the data into frames of 1024 samples with a hop of 512 samples (50 % overlap), which corresponds to windows of roughly 100 ms.

2. Mathematical feature extraction

For every frame, a feature vector is computed from the time-domain and frequency-domain representations. The time-domain features include the root mean square value:

$$\text{RMS} = \sqrt{\left[(1/N) \sum_{i=1}^N x_i^2 \right]} \quad (1)$$

and the kurtosis, which is sensitive to the impulsive content produced by a damaged cutting edge:

$$K = E[(x - \mu)^4] / \sigma^4 \quad (2)$$

In the frequency domain, the magnitude spectrum is obtained with a real FFT (np.fft.rfft) followed by peak detection. The tooth-passing frequency is calculated from the spindle speed and the number of teeth:

$$\text{FTPF} = (\text{RPM} / 60) \times z \quad (3)$$

The software is organised as a modular pipeline (Fig. 4) in which each stage hands its output to the next. Acquisition, preprocessing, feature extraction, and the classification and RUL module run in Python or LabVIEW, and the results are shown on the dashboard

The sideband energy is the sum of the FFT bins around the tooth-passing frequency and its harmonics. For the test setup of Section 6, a spindle speed of 12 000 RPM gives a spindle frequency of 200 Hz and a tooth-passing frequency of 600 Hz for a 3-flute cutter. Together with the peak value, crest factor and spectral centroid, these features allow the subsequent stages to follow the behaviour of the signal rather than only its absolute amplitude.

V. WEAR CLASSIFICATION, RUL ESTIMATION AND DASHBOARD

1. Threshold-based wear classification

A rule-based classifier assigns each frame to one of three states: Healthy, Early-Wear or Critical. The RMS acceleration is compared against calibrated limits, with Early-Wear declared when RMS exceeds 2.5 g and Critical when it exceeds 5 g. The calibration of these thresholds against measured data from the test bench is in progress.

2. Remaining useful life estimation

A polynomial trend is fitted to the RMS feature over successive cuts. The remaining useful life is then estimated by linear extrapolation of the trend slope to the critical threshold:

$$\text{RUL} = (\text{RMS}_{\text{threshold}} - \text{RMS}_{\text{current}}) / \text{trend slope} \quad (4)$$

Exponential smoothing is applied to the feature series before the fit to reduce the effect of cycle-to-cycle variation.

3. System outputs

The software provides three primary outputs for the operator:

- **Real-Time Dashboard:** a LabVIEW front panel (or Tkinter GUI) with six panels: a 3-channel live waveform (1 s window), an FFT spectrum (0–5 kHz, log-scale amplitude in dB), a wear gauge (0–100 % with colour zones), an RUL countdown in minutes or hours to replacement, a current-draw trend with an alarm at more than 15 A sustained, and a timestamped event log.
- **Alerting:** an on-screen alarm and an audio beep, with an SMS or e-mail alert through the Twilio API when the critical state is reached.
- **Data Logging:** automated timestamped archiving of the raw data and the extracted features in CSV format for later analysis and model training.

VI. EXPERIMENTAL SETUP, PROJECT STATUS AND EXPECTED RESULTS

1. Test bench and validation plan

The system is validated on a benchtop CNC router (Genmitsu 3018-Pro) with tools of known condition. A 3-flute end mill ($\varnothing 6$ mm) cuts aluminium 6061 at 12 000 RPM, and four tool conditions are tested: new, 50 % worn, 80 % worn and broken. Sensor placement is validated with a dial gauge and an oscilloscope, and the FFT peaks are verified against the known spindle frequency of 200 Hz. The false alarm rate is evaluated over 100 cycles.

2. Project status

At the time of writing, the project topic selection and literature survey, the component procurement and the KiCad schematic and layout of the signal-conditioning PCB are complete. On the software side, the Python acquisition script with ring buffer, the wavelet denoising module (implemented and unit-tested), the FFT module with Hanning window (verified on a known sine signal) and an initial

LabVIEW front-panel prototype with waveform and FFT panels have been completed.

PCB fabrication and soldering, sensor mounting and wiring on the test bench, the feature extraction module (kurtosis and sideband energy) and the threshold calibration of the wear classifier are in progress. Integration of all modules into a single application, the full wear-stage validation tests, the RUL accuracy benchmarking, the dashboard polish and alert integration, the report and the final demonstration are pending. No experimental accuracy results are therefore reported in this paper; the figures below are design targets.

3. Expected results and outcomes

The system is designed to meet the following targets: a tool wear classification accuracy of at least 90 % across the four wear classes, a false alarm rate below 5 %, an RUL estimation error (RMSE) below 10 % of the actual remaining life, a sampling rate of 10 kSa/s, an alert latency below 100 ms and real-time detection of four wear stages. If these targets are met, the system can support reductions in unplanned CNC downtime and extensions of tool life of the order reported for PdM, with applications in the automotive, aerospace and mould-making industries.

VII. CONCLUSION

Future Work

This paper presents the design of a real-time tool life monitoring system based on virtual instrumentation. By combining a vibration, acoustic emission and spindle current sensor array, a conditioning and acquisition stage, and a LabVIEW/Python pipeline of wavelet denoising, windowed FFT, feature extraction, threshold classification and trend-based RUL estimation, the system aims to provide a low-cost and scalable route to predictive maintenance in automated manufacturing. Hardware and software modules are partly complete, and validation against the four tool-wear conditions is under way.

The following extensions are planned:

- **Machine learning integration:** training CNN or LSTM models on labelled vibration datasets such as the PHM 2010 challenge data for higher accuracy and better generalisation across tool types and materials.
 - **Cloud and IIoT connectivity:** pushing sensor data to AWS IoT Core or Azure IoT Hub via MQTT for remote dashboards and fleet-level analytics.
 - **Wireless sensor network:** replacing wired sensors with Bluetooth or ZigBee nodes on each axis to reduce cabling and allow hot-swap sensor modules.
 - **Thermal imaging fusion:** combining IR thermal camera data with vibration signals to improve the detection of adhesive wear and built-up edge formation.
 - **Multi-machine deployment:** monitoring more than 10 CNC machines through a central SCADA dashboard with anomaly correlation and maintenance scheduling.
 - **Mobile alert app:** a React Native or Flutter application with push notifications, a historical trend viewer and one-tap work-order creation.
7. PHM Society, 2010 PHM Society Conference Data Challenge [Dataset]. phmsociety.org/competition/phm/10
 8. ISO 10816-3:2009, Mechanical vibration – Evaluation of machine vibration by measurements on non-rotating parts – Part 3: Industrial machines
 9. Analog Devices, ADXL345 Digital Accelerometer Datasheet, Rev. E (2015)
 10. Allegro MicroSystems, ACS712 Fully Integrated, Hall Effect-Based Linear Current Sensor Datasheet (2019)
 11. Arduino, Arduino Mega 2560 Reference (Arduino LLC, 2023). arduino.cc
 12. Python Software Foundation, NumPy, SciPy, PyWavelets libraries. numpy.org, scipy.org, pywavelets.readthedocs.io

REFERENCES

1. A.G. Rehorn, J. Jiang, P.E. Orban, State-of-the-art methods for machine tool condition monitoring – a review. *Int. J. Adv. Manuf. Technol.* 34, 363–378 (2007)
2. D.E. Dimla, Sensor signals for tool-wear monitoring in metal cutting operations. *Int. J. Mach. Tools Manuf.* 40(8), 1073–1098 (2000)
3. C. Scheffer, P.S. Heyns, Wear monitoring in turning operations using vibration and strain measurements. *Mech. Syst. Signal Process.* 15(6), 1185–1202 (2001)
4. S. Nandi, H.A. Toliyat, X. Li, Condition monitoring and fault diagnosis of electrical motors – a review. *IEEE Trans. Energy Convers.* 20(4), 719–729 (2005)
5. S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd edn. (Academic Press, 2008)
6. National Instruments, *LabVIEW User Manual* (NI, 2020). ni.com/labview