

A Low-Cost Real-Time Vibration Monitor for a Laboratory Wind-Turbine Rotor: Design, Signal-Chain Verification and the Pitfalls of Single-Window Analysis

Rishi Kinger¹ Atharva Singh²

School of Electrical Engineering Vellore Institute of Technology (VIT) Vellore, India

Abstract— Low-cost MEMS accelerometers and microcontrollers make continuous vibration monitoring affordable for small wind turbines, but published low-cost systems seldom document their signal chain well enough for a reader to judge what the displayed indicators mean. This paper describes a monitor built from an ESP32 microcontroller and an MPU6050 three-axis accelerometer on a motor-driven turbine rotor. The firmware samples at 100 Hz and removes gravity with a first-order recursive high-pass filter (-3 dB at 0.86 Hz); a MATLAB dashboard computes the RMS level, a Welch power spectral density (PSD) and the X–Y cross-correlation over a 10-s sliding window. Each processing stage is characterized analytically, including the 95% confidence interval of the PSD estimate (-3.0 dB to $+4.6$ dB, 11 equivalent degrees of freedom). In one recorded window, the RMS levels were 163, 224 and 170 mg on the X, Y and Z axes, and the integrated Z-axis PSD (166 mg RMS) agreed with the time-domain value within 3%. The Z-axis spectrum peaked at 6.7 Hz, 21 dB above the 12–50 Hz floor. Segment-wise analysis showed that this peak and the oscillation of the X–Y correlation came mainly from a 1.6-s transient, while the less disturbed parts of the record oscillated at 2.8–3.8 Hz, so the single-window indicators mixed transient and steady vibration. The paper also identifies three configuration risks common to such systems — a sensor low-pass filter set above the Nyquist frequency, possible clipping on the gravity-bearing axis and the absence of sample timestamps. The results rest on a single 10-s record without a shaft-speed reference; validated fault detection is not claimed, and the experiments needed to establish it are specified.

Keywords— condition monitoring, ESP32, MEMS accelerometer, MPU6050, non-stationary vibration, power spectral density, small wind turbine, Welch method

I. INTRODUCTION

Most mechanical faults in wind turbines — rotor imbalance, misalignment, loose mountings and bearing wear — change the vibration of the structure before they cause a failure. Vibration-based condition monitoring uses this: a fault adds energy at characteristic frequencies, and a change in the spectrum gives a warning that a fixed maintenance schedule cannot [1], [2]. Utility-scale turbines carry dedicated monitoring systems, but such systems are rarely economical for small turbines used on farms, rooftops and in laboratories [3].

MEMS accelerometers now offer enough resolution and bandwidth for low-frequency machine vibration [4], [5], and inexpensive microcontrollers can sample

them at a fixed rate. Many low-cost monitors have been built from such parts. Their weak point is rarely the hardware. It is the signal chain between the sensor and the number on the screen: the sensor's internal filter, the sampling rate, the filtering applied before analysis, the statistical reliability of a spectral estimate, and the assumption that the analysed window is stationary. When these are not stated, a reader cannot tell whether a spectral peak is a property of the machine or an artefact of the instrument.

This paper presents a monitor built from an ESP32 microcontroller, an MPU6050 accelerometer and a MATLAB dashboard, tested on a small turbine rotor driven by an electric motor. Its contributions are:

- a complete description of the acquisition and analysis chain, with each stage characterized analytically (filter response, spectral resolution, equivalent degrees of freedom and confidence interval of the PSD);
- an internal consistency check of the chain (Parseval agreement between the PSD and the time-domain RMS);
- a segment-wise re-analysis of a recorded dashboard window, showing that the dominant spectral peak and the inter-axis correlation arise from a transient rather than from steady vibration, together with a quantitative test that distinguishes the two;
- an account of configuration risks found in this system and common to similar ones, with the corrections and experiments they require.

The paper does not claim validated fault detection. No shaft-speed reference or seeded fault was available for the recorded data; Section VIII lists the experiments needed before such a claim can be made.

Section II reviews related work. Section III describes the system and Section IV the signal processing and its measurement properties. Section V describes the recorded data, Section VI presents the results and Section VII discusses them. Section VIII covers limitations and future work, and Section IX concludes.

II. RELATED WORK AND RESEARCH GAP

A. Wind-Turbine Condition Monitoring

Reviews of wind-turbine monitoring identify vibration analysis as a widely used technique and cost and data handling as the main barriers to its wider use [1]–[3]. In rotating machines, faults appear at predictable multiples (orders) of the shaft rate: imbalance at the rotation rate (1X), misalignment typically at 2X, blade-pass excitation at the number of blades times the rotation rate, and looseness as many harmonics and sometimes half-orders. A resonance occurs when a forcing frequency

coincides with a natural frequency of the structure [6]. Assigning a peak to a fault therefore requires the shaft speed. Standards such as ISO 20816-1 [7] and, for geared wind turbines, VDI 3834-1 [8] define measurement quantities, positions and evaluation zones for full-scale machines.

B. MEMS Accelerometers for Vibration Measurement Albarbar et al. [4] compared three MEMS accelerometers

with a piezoelectric reference under sinusoidal, random and impact excitation and on a machine tool; two were judged suitable for condition monitoring and one was not, which shows that suitability must be tested rather than assumed. Hassan et al. [5] review vibration sensors for condition monitoring, including studies in which low-cost MEMS accelerometers showed a higher noise density than a piezoelectric reference. Srokosz et al. [9] screened 24 low-cost MEMS accelerometers by Allan variance and placed the MPU6050 in the lowest of three noise grades. For the low frequencies and the accelerations of tens to hundreds of milli-g that matter here, the MPU6050's noise density of 400 $\mu\text{g}/\text{Hz}$ [10] is adequate, but its zero-g offset and sensitivity tolerances (± 50 to ± 80 mg and $\pm 3\%$) are not negligible [10].

C. Low-Cost Monitoring Systems and Their Validation

Table I compares systems that report how low-cost MEMS measurements were checked. Two patterns stand out. Systems validated against a piezoelectric reference [4], [11], [12] report quantitative errors, but they are laboratory characterizations rather than monitors of a running machine. Systems built around the MPU6050 and an ESP32 or Arduino for machine monitoring [13]–[15] demonstrate working prototypes, but none reports a reference measurement or the statistical uncertainty of the spectral indicators it displays.

D. Research Gap

The gap addressed here is therefore not a new sensor or algorithm. It is the lack, in low-cost monitors of this class, of (i) a stated and verified signal chain, including the anti-aliasing condition and the confidence of spectral estimates, and (ii) a

check that the window from which RMS, PSD and correlation are computed is stationary. Both determine whether

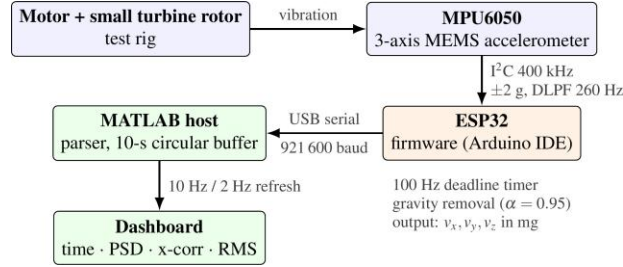


Fig. 1. Signal path of the monitor as used for the recorded data. the indicators on a dashboard can be interpreted as properties of the machine.

III. SYSTEM DESIGN

A. Test Rig

A small turbine rotor was mounted on the shaft of an electric motor, which drove it during the test, and the MPU6050 was attached to the motor–turbine assembly. The motor model, drive setting, rotor geometry, sensor attachment method and sensor axis orientation were not recorded for the test analysed here. Their absence limits the interpretation of the results, as discussed in Sections VII and VIII.

B. Acquisition Hardware

Fig. 1 shows the signal path and Table II the configuration. The MPU6050 combines a three-axis

accelerometer and gyroscope with 16-bit converters and an internal digital low-pass filter (DLPF) [10]. Only the accelerometer is used, on its 2 g range (16 384 LSB/g, about 0.06 mg per count). The ESP32 [16] reads the sensor over I2C at 400 kHz and sends one comma-separated line per sample over USB at 921 600 baud. A line is at most 27 characters, so 100 lines per second use under 3% of the link capacity.

C. Firmware

The firmware (Appendix, Listing 1) uses a deadline scheduler. On each pass the loop compares the microsecond counter with the next deadline and, when it is reached, advances the deadline by exactly 10 000 μ s. Because the deadline is incremented rather than reset to the current time, a late iteration does not shift later samples. The signed comparison of the counter difference remains correct when the 32-bit counter wraps after about 71.6 min. At each sample the firmware reads the sensor, converts each axis from m/s² to g, subtracts a running estimate of the static (gravity) component and transmits the result in mg.

D. Dashboard

The MATLAB program reads one line at a time, discards any line that does not parse into three finite numbers, and writes valid samples into three circular buffers of 1000 samples. Chronological order is restored only when the display is refreshed. Rendering is decoupled from acquisition by three

Table I
Comparison With Low-Cost Mems Vibration Systems Reported In The Literature

Study	Sensor and platform	Sampling	Reference / validation	Test conditions and stated limitations
Albarbar <i>et al.</i> 2008	Three MEMS	Not	Piezoelectric reference; sine,	Shaker and CNC machine at 2400 rpm;
[4]	accelerometers (models withheld)	compared here	random and impact tests	one of three sensors unsuitable
Komarizadehasl <i>et al.</i> 2021 [11]	Five averaged MPU9250, Arduino Due	85 Hz	Two PCB piezoelectric references on a servo-hydraulic shaker	0.5–10 Hz; amplitude errors 0.47–1.55%, frequency error <0.014%; not tested on a real structure
Rossi <i>et al.</i> 2023 [12]	ADXL356 on a rotating blade, Raspberry Pi 4	16 kS/s	Wired PCB piezoelectric reference	0–300 rpm; MAPE <5%, increasing with speed
Srokosz <i>et al.</i> 2024 [9]	24 MEMS screened (MPU6050 included), 8 tested further	100–400 Hz	Allan variance; low-frequency exciter	1–2 Hz; MPU6050 graded lowest and not tested further
Rodrigues <i>et al.</i> 2021 [13]	ADXL345 and MPU6050, Arduino UNO	Not reported	Inclinometer for static offsets; no reference accelerometer	DC motor at 5–13 V; MPU6050 offset shifts up to 0.03 g

Kolok <i>et al.</i> 2025 [14]	MPU6050, ESP32-C6	Not compared here	No reference instrument	One seeded imbalance at fixed speed; RMS, Welch PSD and Isolation Forest
Janumala <i>et al.</i> 2026 [15]	MPU6050 with current sensor, ESP32	Not reported	No reference instrument	DC motor with seeded faults; threshold alarms, qualitative results
This work	MPU6050, ESP32, MATLAB	100 Hz	Analytical chain characterization; Parseval check; no reference instrument	Motor-driven turbine rotor; one 10-s record; no speed reference

- a complete, reproducible description of the acquisition and analysis chain, with each stage characterized analytically (filter response, spectral resolution, equivalent degrees of freedom and confidence interval of the PSD);
- an internal consistency check of the chain (Parseval agreement between the PSD and the time-domain RMS);
- a segment-wise re-analysis of a recorded dashboard window, which shows that the dominant spectral peak and the inter-axis correlation arise from a transient rather than from steady vibration, and a quantitative test that distinguishes the two;
- an account of configuration risks found in this system and common to similar ones, with the corrections and validation experiments they require.
- The paper does not claim validated fault detection. No shaft-speed reference or seeded fault was available for the recorded data, and Section VIII lists the experiments needed before such a claim can be made.

Section II reviews related work. Section III describes the system and Section IV the signal processing and its measurement properties. Section V describes the recorded data, Section VI presents the results and Section VII discusses them. Section VIII covers limitations and required validation, and Section IX concludes.

IV. RELATED WORK AND RESEARCH GAP

A. Wind-turbine condition monitoring

Reviews of wind-turbine monitoring agree that vibration analysis is the most widely used technique and that cost and data handling are the main barriers

to its wider use [1]–[3]. In rotating machines, faults appear at predictable multiples (orders) of the shaft rate: imbalance at the rotation rate (1X), misalignment typically at 2X, blade-pass excitation at the number of blades times the rotation rate, and looseness as many harmonics and sometimes half-orders. A resonance occurs when any forcing frequency coincides with a natural frequency of the structure [6]. Assigning a peak to a fault therefore requires the shaft speed. Standards such as ISO 20816-1 [7] and, for geared wind turbines, VDI 3834-1 [8] define measurement quantities, positions and evaluation zones for full-scale machines.

B. MEMS accelerometers for vibration measurement

Albarbar *et al.* [4] compared three MEMS accelerometers with a piezoelectric reference under sinusoidal, random and impact excitation and on a machine tool; two were judged suitable for condition monitoring and one was not, which shows that suitability must be tested rather than assumed. Hassan *et al.*

[5] review vibration sensors for condition monitoring, including studies in which low-cost MEMS accelerometers showed a higher noise density than a piezoelectric reference. Srokosz *et al.* [9] screened 24 low-cost MEMS accelerometers by Allan variance and placed the MPU6050 in the lowest of threenoise grades. For the low frequencies and accelerations of tens to hundreds of milli-g that matter here, the MPU6050's noise density (400 $\mu\text{g}/\sqrt{\text{Hz}}$ [10]) is adequate, but its offsets and sensitivity tolerances are not negligible (± 50 mg to ± 80 mg and $\pm 3\%$, respectively [10]).

C. Low-cost monitoring systems and their validation

TABLE II
CONFIGURATION USED FOR THE RECORDED DATA

Parameter	Value
Microcontroller	ESP32, firmware built in the Arduino IDE
Sensor	MPU6050 (accelerometer only)
Range and resolution	± 2 g; 16 384 LSB/g
Sensor DLPF setting	260 Hz bandwidth
I ² C bus	GPIO21 (SDA), GPIO22 (SCL), 400 kHz
Sampling rate F_s	100 Hz (10 ms deadline timer)
Gravity removal	First-order recursive, $\alpha = 0.95$
Serial link and format	921 600 baud; ASCII CSV in mg
Host software	MATLAB, serialport interface
Analysis window	$N = 1000$ samples (10 s)

rates: every sample is stored; every 10 samples (10 Hz) the time traces and RMS bars are updated; every 50 samples (2 Hz) the PSD and cross-correlation are recomputed. Samples that arrive during a redraw wait in the serial input buffer.

V. SIGNAL PROCESSING AND MEASUREMENT PROPERTIES

A. Gravity Removal

With $a[n]$ the acceleration in g, the firmware computes

$$d[n] = \alpha d[n-1] + (1 - \alpha) a[n], \quad (1)$$

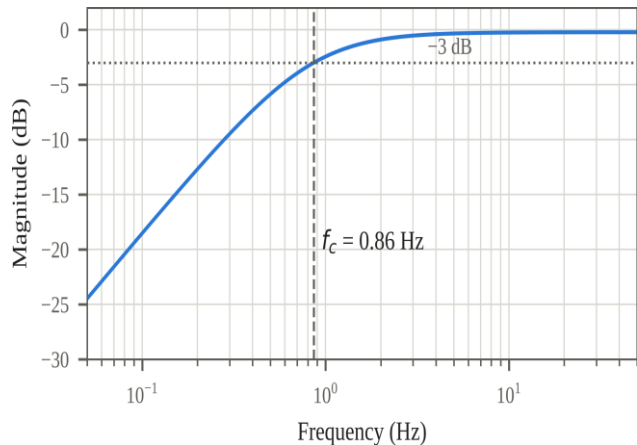
$$v[n] = 1000 a[n] - d[n] \text{ mg}. \quad (2)$$

Combining (1) and (2) gives a first-order high-pass filter

$$H(z) = \frac{\alpha(1 - z^{-1})}{1 - \alpha z^{-1}}, \quad (3)$$

MEMS with a zero at DC and a pole at $z = \alpha$. Setting $|H|_2 = 1/2$ gives the cut-off

$$C = \frac{\max |v[n]|}{V} \quad \cos \omega_c = \frac{1 - 3\alpha^2}{2\alpha - 4\alpha^2} \quad f_c = \frac{\omega_c F_s}{2\pi} \quad (4)$$



The gap this paper addresses is therefore not a Fig. 2. Magnitude response of the gravity-removal filter (3) for $\alpha = 0.95$ at

$F_s = 100$ Hz.

For $\alpha = 0.95$ and $F_s = 100$ Hz, $f_c = 0.86$ Hz and the time constant is $T_s/(1 - \alpha) = 0.2$ s (Fig. 2). Gravity is removed completely, and the attenuation is below 0.5 dB above 3.3 Hz. Because $d[0] = 0$ in the firmware, the output starts with a transient equal to the static reading, which decays to 1% within 90 samples.

B. RMS Level and Crest Factor

For each axis, the dashboard computes the RMS over the N most recent samples,

$$V_{rms} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} v^2[n]}, \quad (5)$$

and the crest factor used in Section VI is

$$C = \frac{\max_n |v[n]|}{V_{rms}}, \quad (6)$$

indicates impulsive content

Table Iii
Processing Parameters And Derived Properties

Stage	Parameter	Value
Gravity removal	Cut-off, time constant	0.86 Hz; 0.2 s
Window	Length	1000 samples (10 s)
Refresh	Time and RMS; PSD and correlation	10 Hz; 2 Hz
Welch PSD	Window; segment length L	Hamming; 256
	Overlap; segments K	50%; 6
	FFT size; bin spacing	512; 0.195 Hz
	Noise bandwidth	0.53 Hz
	Degrees of freedom ν	11.0
	95% interval	-3.0 dB, +4.6 dB
Cross-correlation	Maximum lag	± 200 samples (± 2 s)

C.

Welch Power Spectral Density

The Z-axis PSD is computed by Welch's method [17], directly from the FFT and without toolbox functions. The mean is removed from the window, which is divided into $K = 6$ segments of $L = 256$ samples with 50% overlap ($D = 128$). Each segment is multiplied by a Hamming window $w[n] = 0.54 - 0.46 \cos 2\pi n/(L - 1)$ and zero-padded to $M = 512$ points. With $z_k[n]$ the k -th mean-removed segment,

$$X_k[m] = \sum_{n=0}^{M-1} z_k[n] w[n] e^{-j2\pi mn/M} \quad (7)$$

$$\hat{P}[m] = \frac{cm}{KF_s} \sum_{k=0}^{K-1} |X_k[m]|^2 \quad (8)$$

with $cm = 2$ except at $m = 0$ and $m = M/2$, where $cm = 1$. The six segments cover samples 1–896; the last 104 samples (1.04 s) of the window are not used. Zero-padding gives a bin spacing of 0.195 Hz, but the resolution is set by the window: the equivalent noise bandwidth of a 256-point Hamming window at 100 Hz is 0.53 Hz [18].

Averaging reduces the variance of the estimate, but six overlapping segments are few. The equivalent number of degrees of freedom is [19]

$$\nu = \frac{2K}{1 + 2 \sum_{m=1}^{K-1} \frac{1 - \frac{m}{K} \rho_w^2(m)}{\sum_{m=0}^{K-1} \rho_w^2(m)}}, \quad (9)$$

where

$\rho_w(m) = \sum_n w[n] w[n + mD] / \sum_n w^2[n]$ is the window overlap correlation. For the Hamming window at 50% overlap, $\rho_w(1) = 0.232$ and $\nu = 11.0$. Treating $\hat{v}P/P$ as χ^2 -distributed [19], [20], the true PSD lies within

$$\frac{\hat{v}P}{\chi^2_{\nu, 0.975}} \leq P \leq \frac{\hat{v}P}{\chi^2_{\nu, 0.025}} \quad (10) \quad \text{with } 95\%$$

confidence, which is -3.0 dB to +4.6 dB around the estimate. A local maximum less than about 5 dB above its neighbourhood is therefore weak evidence of a spectral line. Table III lists the processing parameters

D. Cross-Correlation Between Axes

The similarity of the X and Y signals is measured by the normalized cross-correlation, computed with MATLAB's `xcorr(x,y,200,'coeff')`:

$$\rho_{xy}[\ell] = \frac{\sum_n x[n] y[n + \ell]}{\sqrt{\sum_n x^2[n] \sum_n y^2[n]}} \quad |\ell| \leq 200. \quad (11)$$

A periodic component shared by both axes appears as an oscillation of ρ_{xy} with the period of that component. Because the sums in (11) run over $N - \ell$ products but are normalized by full-length energies, a component present throughout the window produces an oscillation whose envelope decreases only linearly with lag,

$$|\rho_{xy}[\ell]| \approx \rho_0 \left(1 - \frac{|\ell|}{N}\right) \quad (12)$$

where ρ_0 is the envelope at zero lag. At the maximum lag of 200 samples the envelope should still be 80% of ρ_0 . An envelope that decays much faster indicates that the shared component lasts only part of the window. This gives a simple stationarity test from a display the dashboard already provides.

E. Sampling and Anti-Aliasing

The sensor's DLPF was set to 260 Hz while the ESP32 sampled at 100 Hz, so the Nyquist frequency was 50 Hz. Vibration between 50 and 260 Hz is not attenuated before sampling and appears in the measured band at

$$f_a = |f - kF_s|, \quad k = \text{round}(f/F_s). \quad (13)$$

For example, a 98 Hz component would appear at 2 Hz and a 130 Hz component at 30 Hz. Motor electromagnetic and bearing vibration can lie in this range, so the measured spectrum cannot be attributed with certainty to content below 50 Hz. The DLPF should be set below the Nyquist frequency (21 or 44 Hz at 100 Hz sampling), or

the sampling rate raised (for example, 500 Hz with a 94 Hz DLPF) [10], [21].

bearing vibration can lie in this range, so the measured spectrum cannot be attributed with certainty to content below 50 Hz. The DLPF should be set below the Nyquist frequency (21 or 44 Hz at 100 Hz sampling), or the sampling rate raised (for example, 500 Hz with a 94 Hz DLPF) [10], [21]. peaks in the recorded window exceeded 1.1 g (Section VI),

carried gravity. In addition, the stream carries no sample index or timestamp. The host assumes a uniform 100 Hz grid, so a lost or malformed line would shift all later samples in time without trace.

VI. RECORDED DATA AND EXTRACTION METHOD

The data analysed here consist of one dashboard window, captured as a screenshot during a test of the motor-driven rotor (Fig. 3). The raw serial stream was not logged, and the operating state of the rig during the window — for example, whether the drive setting changed or the rotor was disturbed — was not recorded

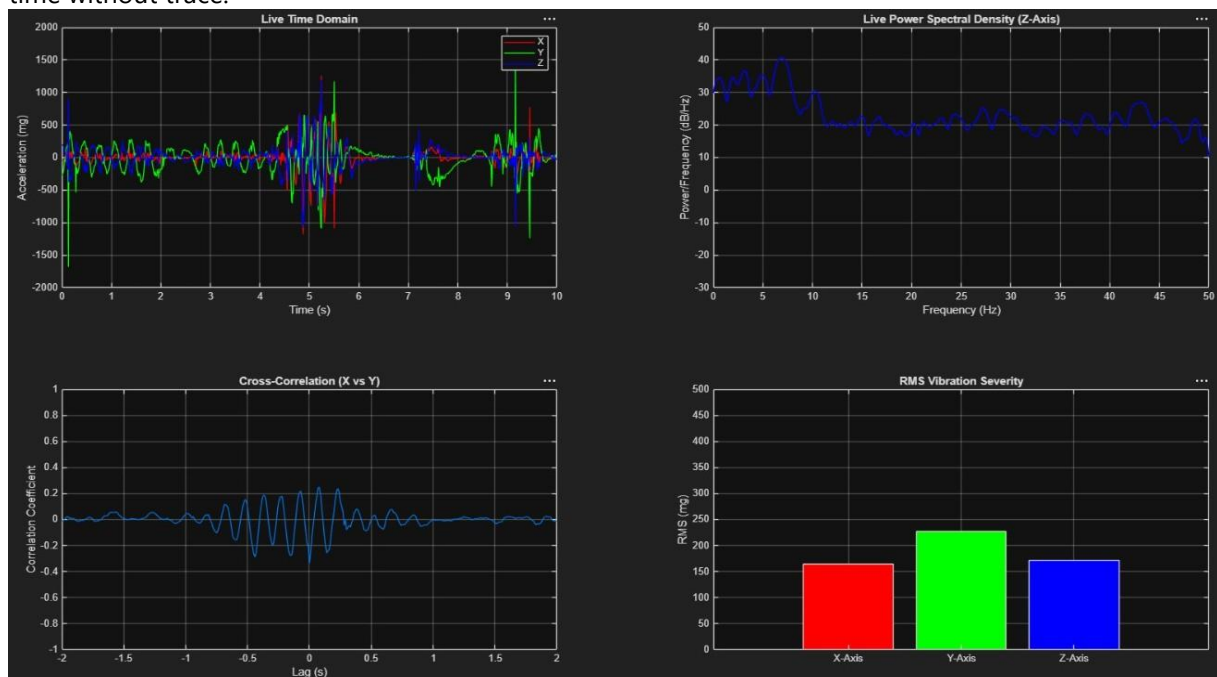


Fig. 3. Dashboard window analysed in this paper (original screenshot, cropped to the plot area). Top left: three-axis acceleration over the last 10 s (X red, Y green, Z blue). Top right: Welch PSD of the Z axis. Bottom left: normalized X–Y cross-correlation over ± 2 s. Bottom right: RMS per axis over the window.

Numerical values were extracted from the screenshot. RMS values were read from the bar heights, and the time traces, the PSD curve and the cross-correlation curve were digitized by locating the plotted line colour in each pixel column and mapping pixel positions to axis values through the grid lines. The read-out resolution is 13.7 mg per pixel for the time

traces, 1.7 mg for the RMS bars, 0.27 dB and 0.093 Hz for the PSD, and 0.0068 and 7.5 ms for the cross-correlation. Where traces overlap, the trace drawn last (Z, then Y) hides the others; the X trace was visible in only 44% of pixel columns and is used only for peak values. These values are therefore estimates with stated read-out uncertainty, not a substitute for raw data.

VII.RESULTS

A. Whole-Window Indicators

Table IV lists the indicators for the 10-s window. The Y axis had the highest RMS level, 224 mg, against 163 and 170 mg on X and Z. All three axes reached peaks above 1 g, and the crest factors of 6.9–7.7 are about five times the value for a sinusoid, which indicates strong impulsive content.

As a check of the analysis chain, the digitized Z-axis PSD was integrated from 0 to 50 Hz. The result, 27 700 mg², corresponds to 166 mg RMS, within 3% of the 170 mg shown by the RMS bar. The agreement confirms the PSD scaling in (8); the small difference is consistent with the read-out uncertainty and with the PSD covering only the first 8.96 s of the window.

Table Iv
Whole-Window Indicators (10 S)

Indicator	X	Y	Z
RMS (mg), ± 3	163	224	170
Positive peak (mg), ± 15	+1253	+1267	+1171
Negative peak (mg), ± 15	-1158	-1678	-1048
Crest factor, ± 0.2	7.7	7.5	6.9

B. Spectrum

Fig. 4 re-plots the Z-axis PSD with its 95% confidence band, and Table V lists the local maxima. The largest, at 6.7 Hz (40.5 dB re 1 mg²/Hz), lies 20.8 dB above the median level from 12 to 50 Hz and 10.6 dB above its immediate neigh-bourhood, well outside the confidence interval. The maximum at 10.1 Hz lies 6.4 dB above its neighbourhood. The maxima at 3.3 and 4.9 Hz lie only 5.8 and 5.2 dB above the median level of the 0.5–5.5 Hz region, close to the +4.6 dB upper confidence limit. That region contains about nine independent spectral estimates, so a chance excursion of this size is plausible, and these two maxima are not treated as resolved lines. Integrating the PSD by band, 36% of the Z-axis power lies below 5 Hz, 48% between 5 and 9 Hz, and 12% above 12 Hz.

The broadband level of about 19.7 dB re 1 mg²/Hz from 12 to 50 Hz is about 28 dB above the sensor's specified noise

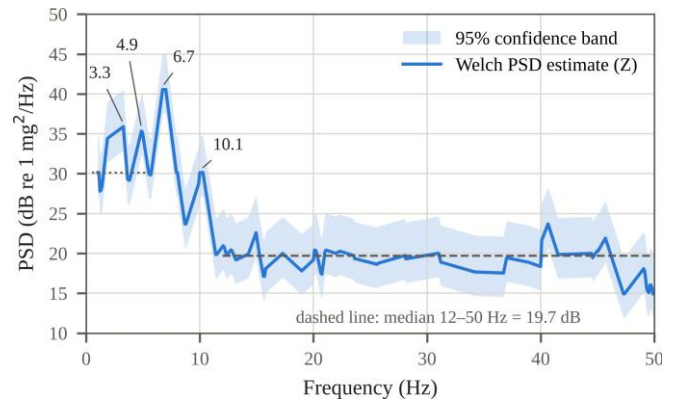


Fig. 4. Z-axis PSD re-plotted from the digitized dashboard curve, with the 95% confidence band of the Welch estimate ($\nu = 11$). Dashed: median level from 12 to 50 Hz; dotted: median level from 0.5 to 5.5 Hz.

Table V
Local Maxima Of The Z-Axis Psd

Frequency (Hz), ± 0.2	Level (dB)	Above 12–50 Hz median (dB)	Above local level (dB)	Assessment
3.3	35.9	16.2	5.8	Marginal
4.9	35.3	15.6	5.2	Marginal
6.7	40.5	20.8	10.6	Resolved
10.1	30.1	10.4	6.4	Probable

density (400 $\mu\text{g}/\sqrt{\text{Hz}}$, or 8 dB re 1 mg²/Hz [10]), so sensor noise does not set the floor of the measured spectrum.

C. Segment-Wise Analysis

The time traces are not stationary (Fig. 5). The window was divided into five segments at visible changes in behaviour, and Table VI gives the largest absolute acceleration, the dominant frequency and the share of signal energy in each. Segment A (0–4.4 s) shows a continuous oscillation at about 2.8 Hz, with peaks of 200–380 mg on Y and 100–250 mg on Z, interrupted by one spike on Y near 0.1 s. Segment B (4.4–6.0 s) is a burst exceeding 1.1 g on all three axes, oscillating at 6.7–6.9 Hz. In segment C (6.0–7.0 s) the signal nearly vanishes: the digitized RMS levels are 10–55 mg, one-third to one-half of those in segment A. Segment D (7.0–8.5 s) begins with a negative excursion on Y that

recovers with a time constant of about 0.3 s, close to the 0.2 s of the gravity-removal filter. Segment E (8.5–10.0 s) contains oscillation near 3.8 Hz with spikes above 1 g. Dominant frequencies were estimated from the digitized Y and Z traces (resampled to 50 Hz, Hann window); the resolution is about 0.6 Hz in segment B and 0.25 Hz in segment A. Energy shares were estimated from the extreme values of the plotted line in each pixel column and are approximate (± 10 percentage points). Segment B, 16% of the window, holds about two-thirds of the Z-axis energy, and its oscillation frequency matches the 6.7 Hz PSD peak. The dominant spectral peak is therefore

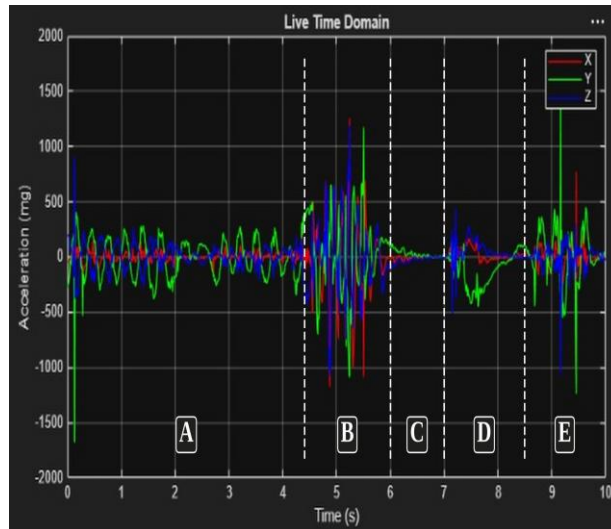


Fig. 5. Time-domain panel of Fig. 3 with the analysis segments A–E marked. The plotted data are unchanged; only the segment labels and boundaries were added.

TABLE VI

SEGMENT-WISE ANALYSIS OF THE RECORDED WINDOW

Seg.	Span (s)	Max $ a $ (mg) X / Y / Z	Dominant freq. (Hz)	Energy Y / Z
A	0–4.4	130 / 1678 / 897	2.8	30% / 18%
B	4.4–6.0	1253 / 1158 / 1171	6.7–6.9	37% / 64%
C	6.0–7.0	34 / 144 / 103	—	<1% / <1%
D	7.0–8.5	158 / 445 / 500	—	9% / 7%
E	8.5–10.0	747 / 1267 / 1048	3.8	24% / 10%

Dominant frequency from the Y and Z traces; — denotes no oscillatory peak above 1.5 Hz.

mainly a property of this transient, not of the steadier vibration in segments A and E.

D. Cross-Correlation

The X–Y cross-correlation (Fig. 6) oscillates with a period of about 0.15 s, which corresponds to 6.6 Hz and agrees with the PSD peak within its resolution. Its envelope is 0.33 at zero lag, falls to 0.16 for lags of 0.6–0.9 s and to 0.06 beyond 0.9 s. For a component present throughout the window, (12) predicts an envelope of 0.26 at 2 s. The observed envelope is about a quarter of that value, so the component shared by the two axes lasted roughly 1–1.5 s, consistent with segment B. The negative value at zero lag indicates that X and Y moved approximately in anti-phase during the burst.

VIII. DISCUSSION

A. What the Record Supports

The monitor works as a measuring instrument in the sense that matters first: its stages behave as designed and its outputs are mutually consistent. The PSD integrates to the RMS level within 3%, the frequency seen in the cross-correlation matches the PSD peak, and the dominant frequencies found in the time traces match the spectral features. The processing in Section IV is standard [17]–[20]; its value here is that each property — cut-off, resolution, confidence interval, stationarity envelope — is stated and checked, which Table I shows is uncommon in comparable low-cost monitors [13]–[15]. or algorithm. It is the lack, in low-cost monitors of this class, of (i) a stated and verified signal chain, including the anti-aliasing condition and the confidence of spectral estimates, and (ii) a check that the window from which RMS, PSD and correlation are computed is stationary. Both determine whether the indicators on a Fig. 6. X–Y cross-correlation re-plotted from the digitized dashboard curve, compared with the envelope (12) expected if the shared 6.6 Hz component were present

throughout the 10-s window.

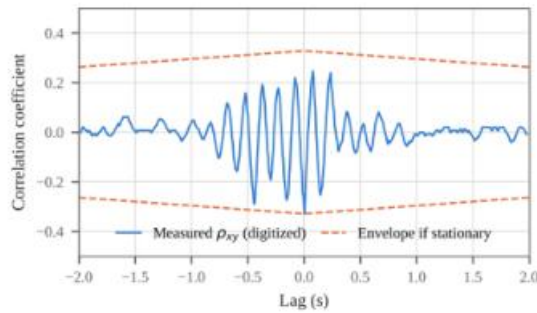


Fig. 6. X–Y cross-correlation re-plotted from the digitized dashboard curve, compared with the envelope (12) expected if the shared 6.6 Hz component were present throughout the 10-s window

B. What the Record Does Not Support

The record does not support claims about the condition of the rotor. Three facts limit it. First, the window is non-stationary: a 1.6-s burst carries most of the energy, the signal nearly vanishes for about 1 s, and segment D begins with an excursion whose recovery resembles the response of the gravity-removal filter to a step in static acceleration, such as a tilt of the sensor or a change in the rig's state. Whole-window RMS, PSD and correlation mix these episodes. Second, without a shaft-speed reference no peak can be assigned to an order; the 6.7 Hz oscillation in segment B may be the ringing of a structural mode excited by an impact or by a change in operating state, and an impact (tap) test would show whether the rig has a natural frequency there [22]. Third, aliasing (Section IV-E) and possible clipping on the gravity-bearing axis (Section IV-F) cannot be excluded for this record.

C. Implications for Low-Cost Dashboards

A dashboard that computes RMS, PSD and correlation from one sliding window presents the average of whatever happened in that window as if it were a steady property. The cross-correlation envelope test (12) and a block-RMS trace, which shows how the RMS level varies over 1-s blocks, cost almost nothing to add and would have flagged this record as non-stationary. Logging raw data with sample indices

allows the same window to be re-analysed with a spectrogram, which separates transient and steady components directly.

IX. LIMITATIONS AND FUTURE WORK

The evidence in this paper comes from a single 10-s window extracted from a screenshot. The raw data, the operating state of the rig during the window, the shaft speed, the rig geometry and the sensor mounting and orientation were not recorded. No reference accelerometer or seeded fault was used, so neither amplitude accuracy nor fault detection has been established. The following experiments, in order of priority, are planned to address these limitations:

- Impact (tap) test: natural frequencies and damping of the rig with the motor off [22].
- Shaft-speed reference: optical or Hall-effect tachometer, so that peaks can be assigned to orders.
- Repeated runs across speeds: at least three 60-s runs at three or more speeds, reported as mean, standard deviation and 95% confidence interval.
- Seeded imbalance: a known small mass at a known radius on one blade, testing whether the 1X amplitude changes beyond run-to-run variation.
- Anti-aliasing comparison: the same condition at 100 Hz with the DLPF at 260 and 21 Hz, and at 500 Hz with the DLPF at 94 Hz.
- Reference measurement: comparison with a calibrated accelerometer at the same mounting point.

Further limitations would remain after these experiments: the rig is motor-driven, so aerodynamic loading and wind-speed effects are not represented, and the evaluation zones of ISO 20816-1 [7] and VDI 3834-1 [8], which apply to full-scale machines and to defined quantities and frequency bands, cannot be applied to this rig directly.

X. CONCLUSION

An ESP32 microcontroller, an MPU6050 accelerometer and a MATLAB dashboard form a working real-time vibration monitor for a small turbine rotor, built entirely from inexpensive commodity parts. Its signal chain was characterized analytically and passed an internal consistency check. Re-analysis of a recorded window showed that the dominant 6.7 Hz spectral peak and the X–Y correlation came mainly from a 1.6-s transient, and that whole-window indicators therefore mixed transient and steady vibration. For low-cost monitors of this kind, the main requirements are not more processing but a stated anti-aliasing condition, raw logging with timestamps, confidence intervals on spectral estimates and a stationarity check. The experiments in Section VIII are required before the monitor can be used to detect faults.

DECLARATION OF AI ASSISTANCE

The authors used Claude (Anthropic), an AI assistant, to draft and restructure the manuscript text, edit its language, locate and check references, digitize values from the dashboard screenshot, and review the source code. The authors reviewed the final manuscript and are responsible for its content.

Data And Code Availability

The data analysed in this study consist of the dashboard window shown in Fig. 3. The firmware and the MATLAB analysis code used to produce it are listed in the Appendix.

APPENDIX: SOURCE CODE

Listing 1 is the ESP32 firmware used for the recorded data (Arduino IDE, Adafruit MPU6050 and Adafruit Unified Sensor libraries). One comment was corrected: the original comment on the sampling interval read “Target: 1000 samples/s”, whereas the 10 000 μ s interval gives 100 samples/s. The code is otherwise unchanged. Listing 2 is the analysis block of the MATLAB dashboard.

Listing 1. ESP32 firmware used for the recorded data.

```
1 #include <Wire.h>
2 #include <Adafruit_MPU6050.h>
3 #include <Adafruit_Sensor.h>
4
5 Adafruit_MPU6050 mpu;
6
7 const int SDA_PIN = 21;
8 const int SCL_PIN = 22;
9
10 const float ALPHA = 0.95f; // DC removal /
    vibration extraction
11
12 float dcX = 0, dcY = 0, dcZ = 0;
13
14 const uint32_t SAMPLE_INTERVAL_US = 10000; // 100
    Hz
15 uint32_t nextSample;
16
17 void setup() {
18   Serial.begin(921600);
19   Wire.begin(SDA_PIN, SCL_PIN);
20   Wire.setClock(400000);
21
22   if (!mpu.begin(0x68, &Wire)) {
23     Serial.println("ERROR: MPU6050 not detected");
24     while (1) delay(10);
25   }
26
27   // Highest accelerometer sensitivity: +/- 2 g
28   mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
29
30   // Gyroscope is not needed for this vibration
    measurement
31   mpu.setGyroRange(MPU6050_RANGE_250_DEG);
32
33   // Configure sensor's internal low-pass filter
34   mpu.setFilterBandwidth(MPU6050_BAND_260_HZ);
```

```

35
36 Serial.println("vib x mg vib y mg vib z mg");
37
38 nextSample = micros();
39 }
40
41 void loop() {
42   uint32_t now = micros();
43
44   if ((int32_t)(now - nextSample) < 0)
45     return;
46
47   nextSample += SAMPLE_INTERVAL_US;
48
49   sensors_event_t accel, gyro, temp;
50   MPU6050.getEvent(&accel, &gyro, &temp);
51
52   // Convert m/s^2 to g
53   float ax = accel.acceleration.x / 9.80665f;
54   float ay = accel.acceleration.y / 9.80665f;
55   float az = accel.acceleration.z / 9.80665f;
56
57   // Track slow/static acceleration and remove it
58   dcX = ALPHA * dcX + (1.0f - ALPHA) * ax;
59   dcY = ALPHA * dcY + (1.0f - ALPHA) * ay;
60   dcZ = ALPHA * dcZ + (1.0f - ALPHA) * az;
61
62   // Residual acceleration emphasizes vibration
63   float vx = (ax - dcX) * 1000.0f;
64   float vy = (ay - dcY) * 1000.0f;
65   float vz = (az - dcZ) * 1000.0f;
66
67   Serial.printf("%.2f, %.2f, %.2f\n", vx, vy, vz);
68
69 }

```

Listing 2. MATLAB dashboard: PSD and cross-correlation block.

```

1 if mod(sample_count,pad_interval) == 0
2
3   % ---- PSD: toolbox-independent Welch method ----
4   z_signal = z - mean(z);

```

REFERENCES

1. Z. Hameed, Y. S. Hong, Y. M. Cho, S. H. Ahn, and C. K. Song, "Condition monitoring and fault detection of wind turbines and related algorithms: A review," *Renew. Sustain. Energy Rev.*, vol. 13, no. 1, pp. 1–39, 2009, doi: 10.1016/j.rser.2007.05.008.
2. P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. A. Tameghe, and G. Ekemb, "Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges," *Energies*, vol. 7, no. 4, pp. 2595–2630, 2014, doi: 10.3390/en7042595.
3. W. Yang, P. J. Tavner, C. J. Crabtree, Y. Feng, and Y. Qiu, "Wind turbine condition monitoring: Technical and commercial challenges," *Wind Energy*, vol. 17, no. 5, pp. 673–693, 2014, doi: 10.1002/we.1508.
4. A. Albarbar, S. Mekid, A. Starr, and R. Pietruszkiewicz, "Suitability of MEMS accelerometers for condition monitoring: An experimental study," *Sensors*, vol. 8, no. 2, pp. 784–799, 2008, doi: 10.3390/s8020784.
5. I. U. Hassan, K. Panduru, and J. Walsh, "An in-depth study of vibration sensors for condition monitoring," *Sensors*, vol. 24, no. 3, Art. no. 740, 2024, doi: 10.3390/s24030740.
6. R. B. Randall, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. Chichester, U.K.: Wiley, 2011.
7. *Mechanical Vibration — Measurement and Evaluation of Machine Vibration — Part 1: General Guidelines*, ISO 20816-1:2016.
8. *Measurement and Evaluation of the Mechanical Vibration of Wind Turbines and Their Components — Wind Turbines with Gearbox*, VDI 3834 Blatt 1, Aug. 2015.
9. P. E. Srokosz, E. Daniszewska, J. Banach, and M. Śmieja, "In-depth analysis of low-cost micro electromechanical system (MEMS) accelerometers in the context of low frequencies and vibration amplitudes," *Sensors*, vol. 24, no. 21, Art. no. 6877, 2024, doi: 10.3390/s24216877.
10. InvenSense Inc., "MPU-6000 and MPU-6050 product specification," Doc. PS-MPU-6000A-00, Rev. 3.4, 2013.
11. S. Komarizadehasl, B. Mobaraki, H. Ma, J.-A. Lozano-Galant, and J. Turmo, "Development of

- a low-cost system for the accurate measurement of structural vibrations," *Sensors*, vol. 21, no. 18, Art. no. 6191, 2021, doi: 10.3390/s21186191.
12. A. Rossi, G. Bocchetta, F. Botta, and A. Scorza, "Accuracy characterization of a MEMS accelerometer for vibration monitoring in a rotating framework," *Appl. Sci.*, vol. 13, no. 8, Art. no. 5070, 2023, doi: 10.3390/app13085070.
13. J. V. O. Rodrigues, M. P. G. Pedroso, F. F. B. Silva, and R. G. Leão Junior, "Performance evaluation of accelerometers ADXL345 and MPU6050 exposed to random vibrational input," *Res., Soc. Dev.*, vol. 10, no. 15, Art. no. e286101523082, 2021, doi: 10.33448/rsd-v10i15.23082.
14. P. Kolok, M. Hodoň, P. Ševčík, L. Hotz, and N. Remy, "Low-cost IoT-based predictive maintenance using vibration," *Sensors*, vol. 25, no. 21, Art. no. 6610, 2025, doi: 10.3390/s25216610.
15. T. Janumala, K. L. Pavan, K. V. Ambareesh, V. Srirama, M. F. Khan,
16. U. K. Keerthi, and S. Vedanth, "Real-time industrial machine health monitoring and failure prediction using ESP32," *Int. J. Eng. Res. Technol.*, vol. 15, no. 6, Paper IJERTV15IS060098, Jun. 2026, doi: 10.5281/zenodo.20579053.
17. Espressif Systems, "ESP32 series datasheet." [Online]. Available: <https://www.espressif.com/en/support/download/documents/chips>
18. P. D. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, 1967, doi: 10.1109/TAU.1967.1161901.
19. F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.
20. D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1993.
21. J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*, 4th ed. Hoboken, NJ, USA: Wiley, 2010.
22. A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
23. D. J. Ewins, *Modal Testing: Theory, Practice and Application*, 2nd ed. Baldock, U.K.: Research Studies Press, 2000.