

Article

A Combined Error-Compensation and Adaptive Third-Order PLL Demodulation Method for TMR-Based Magnetic Encoders

Yue Xin ¹, Jia Cui ^{1,*}, Haifeng Wei ¹ and Li Lui ^{1,2}

¹ School of Automation, Jiangsu University of Science and Technology, Zhenjiang 212000, China; 242210302106@stu.just.edu.cn (Y.X.); whf21@just.edu.cn (H.W.); 231210301211@stu.just.edu.cn (L.L.)

² Suzhou Veichi Electric Co., Ltd., Suzhou 212000, China

* Correspondence: cuijia@just.edu.cn; Tel.: +86-184-5248-2668

Abstract

TMR-based magnetic encoders provide sensitive SIN/COS signals, but practical accuracy is degraded by channel mismatch and decoder dynamics. This study evaluates an end-to-end embedded implementation on a PMSM (Permanent Magnet Synchronous Motor) bench. We consider amplitude mismatch, quadrature non-orthogonality, and harmonic/noise disturbances in the measured differential channels. We implement a lightweight compensation chain, including fixed-window moving-average filtering, min–max amplitude normalization, and correlation-based quadrature identification with sample-shift correction. We then compare four demodulation configurations under identical sampling and reference alignment to a 24-bit encoder: (A0) conventional second-order PLL (phase locked loop), (A1) compensation + open-loop atan2, (A2) compensation + fixed- ω_n third-order PLL, and (A3) compensation + adaptive- ω_n third-order PLL. Experiments with a TMR3081 sensor and an STM32 controller show clear differences among A0–A3. In steady operation, A3 removes the DC bias observed with A0 and keeps the angle error within approximately $\pm 0.3^\circ$ in the evaluated steady window. During commutation and ramp-like segments, PLL-based tracking (A0/A2/A3) is more robust than open-loop atan2 (A1), and bandwidth adaptation in A3 improves the acquisition–noise trade-off within the preset ω_n bounds. These results are reported for this prototype and the tested parameter settings.

Keywords: tunnel magnetoresistance (TMR); amplitude normalization; phase non-orthogonality; harmonic disturbance; adaptive third-order PLL; closed-loop demodulation

1. Introduction

Magnetic encoders serve as essential position-sensing components in modern electromechanical systems, providing angular feedback for servo control, drive commutation, and precision motion tasks. In commercial implementations, magnetic-field sensing is commonly realized using Hall-effect sensors [1], anisotropic magnetoresistance (AMR) sensors [2,3], or giant magnetoresistance (GMR) sensors [4,5]. While Hall sensors are structurally simple and inexpensive, they are typically limited by moderate sensitivity and relatively high power consumption. AMR sensors offer improved sensitivity but may suffer from degraded stability at elevated temperatures. GMR sensors can further enhance sensitivity and thermal characteristics, yet their fabrication complexity and associated costs remain nontrivial [6].

Recent advances in nanoscale materials and device engineering have enabled tunnel magnetoresistance (TMR) sensors, which leverage quantum tunneling effects to achieve



Academic Editor: Matteo Bruno Lodi

Received: 14 January 2026

Revised: 10 February 2026

Accepted: 11 February 2026

Published: 18 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

high sensitivity, low power consumption, and improved temperature stability. Accordingly, TMR sensors have become attractive candidates for next-generation magnetic encoders [7,8].

A typical biaxial TMR angle sensor uses four matched TMR resistive elements arranged as a Wheatstone full bridge. The bridge converts the magnetic-field direction into two differential voltages. These outputs correspond to the field projections on two sensing axes. Under ideal alignment, the two channels form a quadrature SIN/COS pair with an approximately 90° phase separation. This enables full 360° angle reconstruction by arctangent or closed-loop tracking. In practice, the measured SIN/COS pair is rarely ideal. The two differential channels can show amplitude mismatch and DC offset. The phase separation can deviate from 90° due to mechanical misalignment and front-end asymmetry. Harmonic distortion and noise can appear due to magnetic-field harmonics, eccentricity, and electronic noise. Signal processing and acquisition can also introduce effective delay, for example from ADC timing and fixed-window averaging. These nonidealities accumulate and directly limit the achievable angular accuracy in a TMR encoder implementation.

Beyond the sensing element, angular accuracy is strongly influenced by signal nonidealities and decoding algorithms. Prior work has explored multi-sensor fusion strategies to mitigate drift and region-dependent errors [9,10], iterative least-squares identification of stochastic error parameters for inertial/attitude sensors [11], neural-network-based parameter tuning for active disturbance rejection control [12], Kalman-filter-based compensation for improved angle observation in smooth operating regions [13], and iterative virtual-cosine calibration for phase distribution optimization in sinusoidal encoders [14].

In encoder decoding, typical software approaches include direct arctangent demodulation, Coordinate Rotation Digital Computer (CORDIC) methods, look-up-table (LUT) mapping, and phase-locked-loop (PLL)-based closed-loop tracking decoders [15]. The arctangent method is straightforward but may exhibit quadrant-boundary discontinuities and large deviations when signals cross axes [16]. CORDIC reduces trigonometric operations to shift-and-add iterations, achieving high numerical efficiency [17–19], although iterative latency can still affect real-time performance. LUT methods accelerate mapping by storing calibrated voltage–angle pairs in read-only memory (ROM) [20]. PLL-based tracking decoders are widely adopted in noisy environments due to strong disturbance rejection and fast dynamic response [21]. Adaptive-reference PLL designs have been reported for fundamental-error compensation [22–24], and Newton–Raphson method-based three-order PLL variants have been used to obtain accurate rotor position with limited iterations and without manual parameter tuning [25].

Prior work on SIN/COS magnetic encoders can be grouped into: (i) front-end signal acquisition and calibration, and (ii) demodulation/tracking algorithms. Hardware-focused solutions improve signal integrity via offset mitigation, programmable gain, and high-speed ADCs (e.g., SNDR-level and acquisition-distance improvements) but usually leave the dynamic tracking problem to external decoding algorithms.

Calibration-centric methods systematically analyze amplitude mismatch, offset, and non-orthogonality, and provide normalization/non-orthogonality correction procedures; however, they are mostly open-loop and do not address PLL tracking dynamics under acceleration and commutation.

On the decoding side, third-order PLL/DPLL-based designs have been proposed to eliminate steady-state error under ramp inputs and reject disturbances, sometimes coupled with adaptive schemes; nevertheless, most studies are built around general magnetic encoders or specific absolute encoder structures and are not integrated with a lightweight, embedded-friendly front-end compensation pipeline tailored to TMR SIN/COS signals under realistic motor commutation and ramp conditions.

Therefore, as summarized in Table 1, a remaining gap is an end-to-end framework that jointly addresses TMR front-end nonidealities (amplitude mismatch, non-orthogonality, harmonics/noise, delay) and closed-loop demodulation dynamics with a fair second- vs. third-order PLL comparison under realistic PMSM commutation and dynamic tests.

Table 1. Quantitative comparison of representative SIN/COS magnetic encoder signal-conditioning and PLL-based demodulation methods.

Ref.	Sensor/Scenario	Error Sources	Demodulation/Decoder	Adaptation
Silva et al. (2024) [22]	TMR sensor + magnetic scale for industrial positioning	Bridge offset, low-level signal acquisition, robustness vs. vibration/contamination	Outputs SIN/COS for subsequent position algorithms	Not focused
Allegro AN126 (2023) [26]	2D magnetic sensors SIN/COS angle computation	Amplitude mismatch, offset, non-orthogonality; shows angular error harmonic content	ArcTan-based angle (open-loop)	Not focused
Tran et al. (2019) [27]	Absolute magnetic encoder with quadrature signals	Offsets, amplitude mismatch, harmonics, noise affecting PD	Dual PLL designed to keep zero steady-state error under phase jump	HR + DPLL (adaptive through gradient)
Nguyen et al. (2019) [23]	Absolute magnetic encoders; disturbed SIN/COS	Amplitude differences, offsets, phase shift, harmonics, noise	PLL via dominant pole approximation; eliminates DC error under phase/freq/ramp	Adaptive (ALN + PLL)
Smadi et al. (2023) [21]	PLL under distorted grid conditions (not encoder)	DC offset + harmonics under non-ideal input	Third-order PLL, improved dynamic response	Yes (design optimization)
Kuntz et al. (2022) [28]	SIN/COS encoder error theory	Rigorous harmonic analysis of arctan angular error with Fourier series distortions	ArcTan error analysis	Not focused
This work (2026)	TMR, commutation and dynamic tests	Amplitude mismatch, phase non-orthogonality, harmonics/noise, propagation delay + decoder dynamics	Adaptive natural-frequency third-order PLL, compared with 2nd-order baseline	Yes (adaptive ω_n)

This paper makes three primary contributions:

1. A systematic error-source analysis for TMR magnetic encoders is presented, covering amplitude mismatch, phase non-orthogonality, harmonic disturbances, and propagation delay.
2. A combined compensation pipeline is proposed, integrating moving-average filtering, amplitude normalization, phase-orthogonality correction via sample shifting, and parameter-based delay compensation.
3. An adaptive natural-frequency third-order PLL tracking demodulator is developed to eliminate DC steady-state error under frequency-ramp conditions while preserving stability and transient performance.

2. Materials and Methods

2.1. TMR Sensor and Differential Signal Model

A biaxial full-bridge TMR angle sensor typically comprises four matched TMR resistive elements arranged as a Wheatstone bridge (Figure 1). During magnet rotation, element

resistances vary with the magnetic-field direction, producing two orthogonal differential outputs corresponding to the magnetic-field projections along the sensing axes (Figure 2). Under ideal conditions, the outputs can be represented as two sinusoidal channels in quadrature (i.e., $\sim 90^\circ$ phase separation), enabling full 360° angle reconstruction.

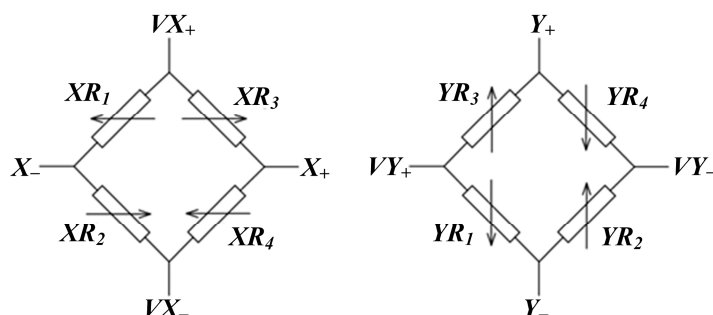


Figure 1. Biaxial full-bridge TMR sensor configuration.

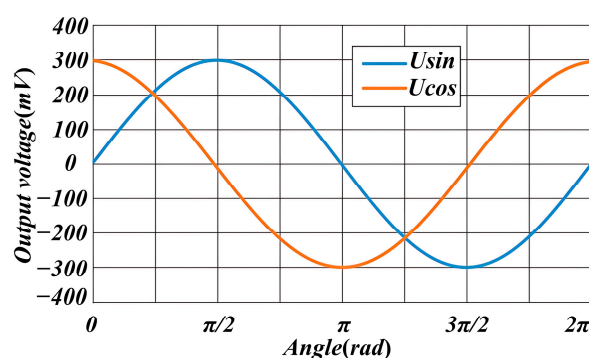


Figure 2. Ideal orthogonal output waveforms of a TMR angle sensor.

Assume four identical TMR elements with an initial resistance R_0 , a resistance variation ΔR induced by the magnetic field, and a supply voltage V_s . The output voltage of the positive half-bridge can be expressed as:

$$V_+ = (R + \Delta R)/2 \cdot R \cdot V_s, \quad (1)$$

The output voltage of the negative half-bridge is:

$$V_- = (R - \Delta R)/2 \cdot R \cdot V_s, \quad (2)$$

By subtracting Equation (2) from Equation (1), the differential output of the Wheatstone full bridge is obtained:

$$V = \Delta R/R \cdot V_s, \quad (3)$$

According to Equation (3), a TMR angle sensor can simultaneously detect the projections of the magnetic field on the sensing plane using elements arranged along the X and Y axes. This configuration ensures two mutually orthogonal cosine-like signals at the output, as shown in Figure 2. By accurate acquisition and demodulation of these signals, the corresponding angular information can be computed, enabling precise measurements over the full 360° range.

2.2. Error Source I: Differential-Channel Amplitude Mismatch

Assume the differential outputs of the TMR chip exhibit an amplitude mismatch characterized by an error coefficient ε_{amp} (within a specified range in the original manuscript). The differential outputs can then be written as:

$$U_{sin} = A \sin \theta \quad (4)$$

$$U_{cos} = (1 + \alpha) A \cos \theta \quad (5)$$

Let A denote the ideal differential amplitude without error, and θ the actual angle. The output angle can be computed via the arctangent method:

$$\varphi = \tan^{-1} \left(\frac{A \sin \theta}{(1 + \alpha) A \cos \theta} \right) = \tan^{-1} \left(\frac{\tan \theta}{1 + \alpha} \right) \quad (6)$$

The angular error due to amplitude mismatch is:

$$\varepsilon_{amp} = \theta - \varphi = \theta - \tan^{-1} \left(\frac{\tan \theta}{1 + \alpha} \right) \quad (7)$$

Taking the tangent on both sides yields:

$$\tan \varepsilon_{amp} = \tan \left(\theta - \tan^{-1} \left(\frac{\tan \theta}{1 + \alpha} \right) \right) \quad (8)$$

Using the tangent difference identity $\tan(a - b) = \frac{\tan a - \tan b}{1 + \tan a \tan b}$, Equation (8) can be simplified to:

$$\tan \left(\theta - \tan^{-1} \left(\frac{\tan \theta}{1 + \alpha} \right) \right) = \frac{\tan \theta - \frac{\tan \theta}{1 + \alpha}}{1 + \tan \theta \frac{\tan \theta}{1 + \alpha}} = \frac{1}{2} \frac{\alpha \sin 2\theta}{1 + \alpha \cos^2 \theta} \quad (9)$$

When the amplitude mismatch is small, $1 + \alpha \cos^2 \theta \approx 1$ and the resulting angular error is also small, $\tan \varepsilon_{amp} \approx \varepsilon_{amp}$, Equation (8) can be further approximated as:

$$\varepsilon_{amp} \approx \frac{1}{2} \alpha \sin 2\theta \quad (10)$$

From Equation (10), the angular error varies sinusoidally with the actual angle, with a frequency twice the mechanical rotation frequency. Hence, for each complete shaft revolution, the error undergoes two full sinusoidal cycles. Moreover, the error amplitude is proportional to the input signal amplitude mismatch. This relationship is instrumental for error modeling and prediction. MATLAB 2024a simulations (with an amplitude mismatch set to 0.1) produce the error curve shown in Figure 3.

Validity and approximation error. It is worth clarifying that the transition from Equation (8) to Equation (9) uses only the standard tangent-difference identity and is therefore an algebraic reformulation rather than an approximation. The first approximation is introduced when simplifying Equation (9) to Equation (10), where (i) the denominator term $1 + \alpha \cos 2\theta$ is replaced by 1 (small mismatch assumption), and (ii) $\tan(\varepsilon_{amp}) \approx \varepsilon_{amp}$ is adopted (small-angle assumption). These approximations are valid when $|\alpha| \ll 1$ and $|\varepsilon_{amp}| \ll 1$ rad, which is consistent with typical gain-mismatch levels reported for SIN/COS magnetic sensors and the corresponding second-harmonic error behavior under small deviations.

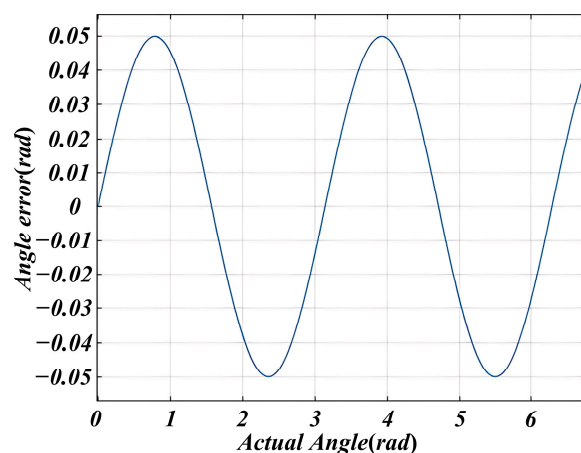


Figure 3. Simulated angular deviation induced by differential-channel amplitude mismatch, $\alpha = 0.1$ (10% amplitude mismatch). (Note: We set $\alpha = 0.1$ (10% gain mismatch) and $\beta = 0.1$ rad ($\approx 5.7^\circ$) as conservative values exceeding the typical TMR3081 synchronization (95–105%) and phase error (87–93°) specifications, to emulate worst-case tolerance and misalignment conditions).

To quantify the approximation accuracy, we numerically compared the exact expression implied by Equation (9), $\varepsilon_{amp,exact}(\theta) = \frac{\arctan(0.5\alpha\sin(2\theta))}{1+\alpha\cos 2\theta}$, with the simplified form Equation (10), $\varepsilon_{amp,approx}(\theta) = 0.5\alpha\sin(2\theta)$. For $\alpha = 0.1$ (10% mismatch), the maximum absolute deviation between $\varepsilon_{amp,exact}(\theta)$ and $\varepsilon_{amp,approx}(\theta)$ over $\theta \in [0, 2\pi]$ is 2.67×10^{-3} rad ($\approx 0.153^\circ$), while the peak error amplitude itself is about 0.05 rad ($\approx 2.86^\circ$). Hence, Equation (10) provides an accurate first-order description of the twice-frequency error component and is used here for insight and parameter-scaling analysis.

The simulation indicates that the angular error reaches peaks at specific angular positions due to unequal amplitudes. Therefore, appropriate measures should be taken to eliminate errors induced by amplitude mismatch in TMR sensor outputs.

2.3. Error Source II: Phase Non-Orthogonality

Let the phase difference between the two differential outputs be β .

$$U_{sin} = A \sin \theta \quad (11)$$

$$U_{cos} = A \cos(\beta + \theta) \quad (12)$$

With differential amplitude A and actual angle θ , the output angle φ can similarly be estimated using the arctangent method. The phase non-orthogonality error is:

$$\varepsilon_{phase} = \theta - \varphi = \theta - \tan^{-1} \left(\frac{\sin \theta}{\cos(\beta + \theta)} \right) \quad (13)$$

Taking the tangent and simplifying yields:

$$\begin{aligned} \tan \varepsilon_{phase} &= \tan \left(\theta - \tan^{-1} \left(\frac{\sin \theta}{\cos(\beta + \theta)} \right) \right) = \frac{\tan \theta - \frac{\sin \theta}{\cos(\beta + \theta)}}{1 + \tan \theta \frac{\sin \theta}{\cos(\beta + \theta)}} \\ &= \frac{\sin \theta \cos \theta \cos \beta - \sin^2 \theta \sin \beta - \sin \theta \cos \theta}{\sin^2 \theta + \cos^2 \theta \cos \beta - \cos \theta \sin \beta \sin \theta} \end{aligned} \quad (14)$$

For small phase deviation and small angular error, there are $\sin \beta \approx \beta$, $\cos \beta \approx 1$, $\tan \varepsilon_{phase} \approx \varepsilon_{phase}$, the expression can be further reduced to:

$$\varepsilon_{phase} \approx \frac{\sin \theta \cos \theta - \beta \sin^2 \theta - \sin \theta \cos \theta}{\sin^2 \theta + \cos^2 \theta - \beta \cos \theta \sin \theta} \approx \frac{-\beta \frac{1 - \cos 2\theta}{2}}{1 - \beta \cos \theta \sin \theta} \approx \frac{1}{2} \beta (\cos 2\theta - 1) \quad (15)$$

Equation (15) shows that the angular error varies approximately as a cosine function of the actual angle, again at twice the mechanical rotation frequency. The error amplitude is proportional to the phase deviation between the two differential signals; as the phase deviation increases, the maximum deviation increases accordingly. In addition, the error baseline shifts downward, resulting in a negative offset relative to the original baseline. MATLAB simulations with a phase deviation of 0.1 generate the error curve shown in Figure 4.

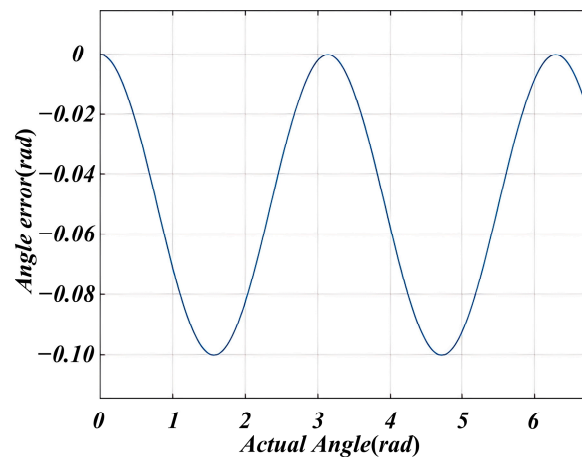


Figure 4. Simulated angular deviation induced by phase non-orthogonality, $\beta = 0.1$ rad ($\approx 5.7^\circ$).

Validity and approximation error (Equations (13)–(15)). The reduction leading to Equation (15) relies on the small-deviation assumptions $\sin \beta \approx \beta$, $\cos \beta \approx 1$, and $\tan \varepsilon_{\text{phase}} \approx \varepsilon_{\text{phase}}$, i.e., $|\beta| \ll 1$ rad and $|\varepsilon_{\text{phase}}| \ll 1$ rad. Under these conditions, the phase non-orthogonality error is dominated by a second-harmonic (2θ) component plus a small DC offset shift, which is consistent with prior analyses of quadrature-signal nonorthogonality in magnetic angle sensors.

We further quantified the approximation accuracy by comparing the exact definition in Equation (13), $\varepsilon_{\text{phase,exact}}(\theta) = \theta - \arctan\left(\frac{\sin \theta}{\cos(\theta + \beta)}\right)$, with the first-order approximation in Equation (15). For $\beta = 0.1$ rad ($\approx 5.7^\circ$), the maximum absolute difference between $\varepsilon_{\text{phase,exact}}$ and $\varepsilon_{\text{phase,approx}}$ over $\theta \in [0, 2\pi)$ is 5.65×10^{-3} rad ($\approx 0.324^\circ$). This confirms that Equation (15) is a valid small-angle approximation for explaining the dominant error mechanism and its scaling with β , while the subsequent compensation is validated experimentally.

The simulation suggests that phase non-orthogonality produces more pronounced errors than amplitude mismatch, making it a critical contributor to the overall error budget of magnetic encoders.

2.4. Error Source III: Harmonic Disturbances

In modern industrial applications and control systems, magnetic encoders are widely adopted as key position sensors to enable precise positioning of mechanical components. During signal acquisition and transmission, magnetic encoder outputs inevitably suffer from harmonic interference originating from diverse sources, including magnetic-field harmonics, thermal noise generated by electronic circuits, and higher-order harmonics induced by misalignment between the magnet center and the shaft center. These disturbances may significantly degrade signal purity and measurement accuracy.

Using Fourier series, the two TMR output channels can be expressed as:

$$U_{\sin} = \sum_{n=1}^{\infty} U_n \sin(n\theta) \quad (16)$$

$$U_{cos} = \sum_{n=1}^{\infty} U_n \cos(n\theta) \quad (17)$$

where U_n denotes the amplitude of the n -th harmonic. Using the arctangent method, the angular error can be obtained as:

$$\varepsilon_h = \theta - \varphi = \theta - \tan^{-1} \left(\frac{\sum_{n=1}^{\infty} U_n \sin(n\theta)}{\sum_{n=1}^{\infty} U_n \cos(n\theta)} \right) \quad (18)$$

$$\tan \varepsilon_h = \tan \left(\theta - \tan^{-1} \left(\frac{\sum_{n=1}^{\infty} U_n \sin(n\theta)}{\sum_{n=1}^{\infty} U_n \cos(n\theta)} \right) \right) = \frac{-\sum_{n=2}^{\infty} U_n \sin((n-1)\theta)}{U_1 - \sum_{n=1}^{\infty} U_n \cos((n-1)\theta)} \quad (19)$$

Under typical conditions $U_n \ll 1$, $U_1 - \sum_{n=1}^{\infty} U_n \cos((n-1)\theta) \approx 1$, higher-order harmonics (above the second) are relatively small, so $\tan \varepsilon_h \approx \varepsilon_h$, allowing further approximation:

$$\varepsilon_h = -\sum_{n=2}^{\infty} \frac{U_n}{U_1} \sin((n-1)\theta) \quad (20)$$

This indicates that the overall angular error can be viewed as a superposition of sinusoidal components; the frequency of each harmonic-induced component scales with $(n-1)$ times the mechanical rotation frequency, and the associated error amplitude is proportional to the harmonic amplitude. MATLAB simulations using the 2nd–5th harmonics (with amplitudes set to 5%, 4%, 3%, and 2% of the fundamental, respectively) yield the composite error curve in Figure 5.

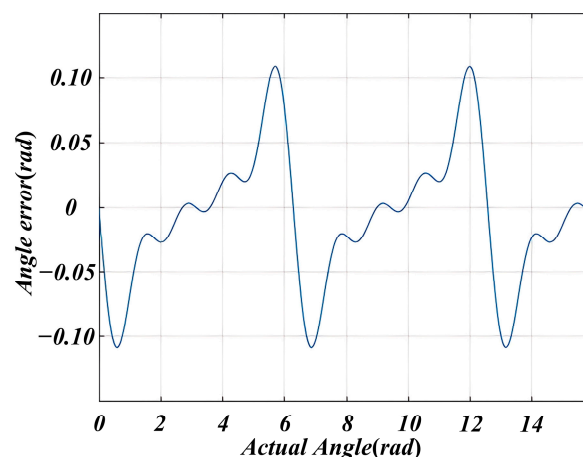


Figure 5. Simulated angular deviation induced by composite harmonic disturbances.

Validity conditions for the harmonic approximation (Equations (19) and (20)). The approximation from Equation (19) to Equation (20) assumes (i) a small distortion regime where the fundamental component dominates, and (ii) a small-angle regime where $\tan \varepsilon_h \approx \varepsilon_h$. More specifically, letting $r_n = U_n/U_1$, denote the relative amplitude of the n -th harmonic ($n \geq 2$), the denominator term in Equation (19) can be written as $1 - \sum_{n=2}^{\infty} r_n \cos((n-1)\theta)$. A sufficient condition for safely approximating this denominator by unity is $\sum_{n=2}^{\infty} r_n \ll 1$, which guarantees that the denominator stays bounded away from zero for all θ and that the resulting error expression is well captured by the first-order term in r_n . This “small-distortion Taylor” viewpoint is consistent with established analyses of arctangent-based angular error under Fourier-series distortions.

In our simulation settings (2nd–5th harmonics with $r_2 = 0.05$, $r_3 = 0.04$, $r_4 = 0.03$, $r_5 = 0.02$), the total distortion level is $\sum r_n = 0.14$, implying the denominator remains within $[1 - 0.14, 1 + 0.14] = [0.86, 1.14]$. Therefore, replacing it by 1 introduces at most an $O(\sum r_n)$ scaling error (worst-case $\lesssim 0.14/0.86 \approx 16.3\%$) in the first-order estimate, while preserving

the correct harmonic orders and phase dependence of the angular error components in Equation (20). Consequently, Equation (20) is used as a compact first-order model to reveal how each harmonic produces an $(n - 1)$ -fold error component and how the error scales with m , whereas the overall compensation-plus-PLL framework is evaluated with measured signals in Section 4.

The results show that under high-order harmonic conditions, the peak positions of the angular error are not fixed but shift with the amplitudes of individual harmonics.

2.5. Combined Compensation Pipeline

2.5.1. Fixed-Window Moving-Average Filtering

A fixed-window moving average filter is an effective approach for smoothing data and reducing noise. The algorithm maintains a window containing the most recent n samples and outputs their mean value. As shown in Figure 6, implementation typically proceeds as follows:

1. Initialization: set the window size (N), clear the buffer, initialize $SUM = 0$, and set $index = 0$.
2. New data arrival: when a new sample data X is received, enter the update loop.
3. Update running sum: update SUM by adding the new sample and removing the value currently stored at $buffer[index]$ (the oldest sample to be overwritten): $SUM = SUM + X - buffer[index]$.
4. Replace buffer element: store the new sample into the buffer at the current position: $buffer[index] = X$.
5. Compute average: output the current moving average: $avg = SUM/N$.
6. Advance index: increment the index for the next update: $index++$.
7. Wrap-around check: if index reaches the window size, reset it to zero to maintain the circular buffer.
8. Repeat: return to the “New data” step for the next sample.

In this design, $N = 32$, so the division in the averaging step can be implemented efficiently as a right shift by 5 bits ($avg = SUM \gg 5$) after the sum update. Each new sample triggers an immediate update with $O(1)$ time per sample and $O(N)$ fixed buffer space, making it suitable for motor-control applications with strict real-time constraints.

Frequency response and bandwidth: The N -point moving-average (MA) filter is a length- N FIR “boxcar” filter with impulse response $h[n] = 1/N$ for $n = 0, \dots, N - 1$ and $h[n] = 0$ otherwise. Its discrete-time frequency response is $H(e^{j\omega}) = \frac{1}{N} \sum_{m=0}^{N-1} e^{-j\omega m} = \frac{1}{N} \cdot \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} = e^{-j\omega \frac{(N-1)}{2}} \cdot \frac{\sin(N\omega/2)}{N \sin(\omega/2)}$, which has a sinc-shaped low-pass magnitude response with periodic spectral nulls. In particular, the first null occurs at $f_1 = f_s/N$, providing a convenient indicator of the filter transition to strong attenuation. For settings ($f_s = 100$ kHz, $N = 32$), the first null is $f_1 = 100,000/32 = 3125$ Hz. In addition, the -3 dB cutoff of an MA filter is well-approximated by $f_{-3dB} \approx 0.443 f_s/N$, which yields $f_{-3dB} \approx 0.443 \times 3125 \approx 1.38$ kHz for $N = 32$ at $f_s = 100$ kHz. Therefore, the MA stage effectively suppresses high-frequency noise and harmonics above the kHz range while preserving the low-frequency SIN/COS fundamental component used for angle estimation.

Group delay and time-domain dynamic response: Because the MA filter is linear phase, it introduces a constant group delay equal to half its order: $\tau_g = \frac{(N-1)}{2}$ samples = $\frac{(N-1)}{2f_s}$ seconds.

With $N = 32$ and $f_s = 100$ kHz, $\tau_g = 15.5$ samples, i.e., $\tau_g = \frac{15.5}{100,000} \approx 155$ μ s. In the time domain, the MA filter’s step response ramps to its final value over approximately N samples, implying an effective settling time on the order of $N/f_s = 32/100,000 = 320$ μ s. These values quantify the latency introduced by the fixed-window averaging stage.

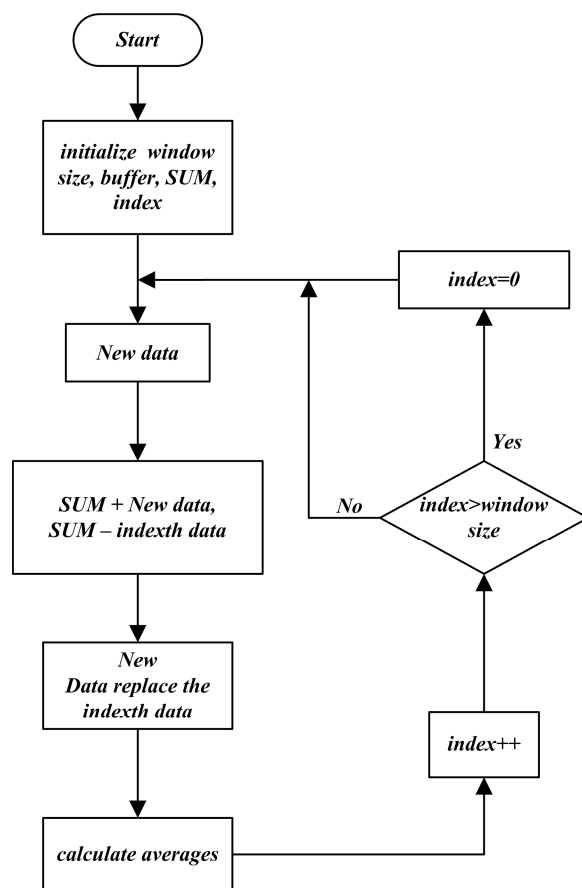


Figure 6. Flowchart of the fixed-window moving-average filter implementation ($N = 32$) using a circular buffer.

Impact on PLL-based tracking (phase lag and stability margin): When placed ahead of the PLL demodulator, the MA filter delay contributes additional phase lag around the PLL crossover region. A practical approximation is that, within the low-frequency passband where the magnitude is near unity, the MA filter behaves like a pure delay of τ_g , thus adding phase lag $\Delta\varphi(\omega) \approx -\omega\tau_g$.

With $\tau_g \approx 155 \mu\text{s}$, the added lag is about -11.2° at 200 Hz ($\omega = 2\pi \cdot 200$), -28.0° at 500 Hz, and -55.8° at 1 kHz. Consequently, the MA window size must be chosen jointly with the PLL bandwidth: increasing f_s (or reducing N) decreases τ_g and improves tracking dynamics, while larger N increases noise suppression but reduces phase margin and can slow acquisition or increase overshoot. This trade-off is consistent with established design guidance for moving-average-filter-based PLL structures, where the MAF window length (hence delay) directly impacts the achievable loop bandwidth and stability margins. In our implementation, $N = 32$ at $f_s = 100 \text{ kHz}$ provides meaningful noise attenuation ($f_{-3\text{dB}} \approx 1.38 \text{ kHz}$) while keeping the fixed latency modest ($\tau_g \approx 155 \mu\text{s}$), which is compatible with stable PLL tracking in the targeted operating conditions.

2.5.2. Amplitude Normalization

Normalization maps a sequence of data to a predetermined numerical range while preserving relative relationships among data points, although global statistical properties may change. Linear normalization is particularly important for improving the measurement accuracy and reliability of magnetic encoders.

For two sinusoidal channels:

$$u_A = U_A \sin \varphi_A + U_0 \quad (21)$$

$$u_B = U_B \sin \varphi_B + U_0 \quad (22)$$

where U_A and U_B are the amplitudes of phase-A and phase-B signals, and U_0 denotes the DC component.

The linear normalization formula is:

$$y = \frac{(Y_{\max} - Y_{\min}) \times (x - X_{\min})}{X_{\max} - X_{\min}} + Y_{\min} \quad (23)$$

where $Y_{\max}$ and $Y_{\min}$ are the upper and lower bounds of the normalization interval; $X_{\max}$ and $X_{\min}$ are the upper and lower bounds of the original data; x is the original data; and y is the normalized output. Equation (23) is the standard min-max normalization (feature scaling) widely used in signal processing and data preprocessing [29].

2.5.3. Phase Orthogonality Identification and Sample-Shift Compensation

The phase difference between the two TMR output channels is first computed to determine whether the two phase signals are orthogonal. Assume the two signals are:

$$u_{\sin}(t) = A \sin \omega t \quad (24)$$

$$u_{\cos}(t) = A \sin(\omega t + \varphi) \quad (25)$$

where A is the signal amplitude, $\omega = 2\pi f$ is the angular speed and f is the signal frequency.

By expanding and manipulating the expressions Equation (25) the phase difference can be obtained.

$$u_{\cos}(t) = A \cos \varphi \sin \omega t + A \sin \varphi \cos \omega t \quad (26)$$

Multiply both sides of Equation (26) by $\cos \omega t$, and let $a = A \cos \varphi$, $b = A \sin \varphi$, the original equation becomes:

$$\cos \omega t \cdot u_{\cos}(t) = a \cos \omega t \sin \omega t + b \cos^2 \omega t \quad (27)$$

Then integrate both sides of Equation (27) and simplify to obtain:

$$\begin{aligned} \int_{t_1}^{t_1+T} \cos \omega t \cdot u_{\cos}(t) dt &= \int_{t_1}^{t_1+T} a \cos \omega t \sin \omega t dt + \int_{t_1}^{t_1+T} b \cos^2 \omega t dt \\ &= \frac{a}{2} \left(\frac{-\cos 2\omega t_1 \cos 2\omega T + \sin 2\omega t_1 \sin 2\omega T + \cos 2\omega t_1}{2\omega} \right) \\ &\quad + \frac{b}{2} \left(T + \frac{\sin 2\omega t_1 \cos 2\omega T + \cos 2\omega t_1 \sin 2\omega T - \sin 2\omega t_1}{2\omega} \right) \end{aligned} \quad (28)$$

where t_1 is any instant, T is the signal period, so $\sin(2\omega T) = 0$, $\cos(2\omega T) = 1$, then Equation (28) directly simplifies to:

$$b = \frac{2}{T} \int_{t_1}^{t_1+T} \cos \omega t \cdot u_{\cos}(t) dt \quad (29)$$

Similarly, multiplying both sides of Equation (26) by $\sin \omega t$ also yields:

$$a = \frac{2}{T} \int_{t_1}^{t_1+T} \sin \omega t \cdot u_{\cos}(t) dt \quad (30)$$

Assume the sampling period is ΔT , let $N = T/\Delta T$, discretizing Equations (29) and (30) gives:

$$a = \frac{2}{N} \sum_{n=0}^{N-1} \sin(n\omega\Delta T) \cdot u_{\cos}(n\Delta T) \quad (31)$$

$$b = \frac{2}{N} \sum_{n=0}^{N-1} \cos(n\omega\Delta T) \cdot u_{\cos}(n\Delta) \quad (32)$$

Because the phase difference φ can be expressed as:

$$\varphi = \tan^{-1} \frac{A \sin \varphi}{A \cos \varphi} \quad (33)$$

By fixing the sampling period during testing and selecting an appropriate parameter in advance, the phase difference can be computed to determine whether it equals $\pi/2$. If not, the required compensation angle can be derived.

After obtaining the phase difference, phase-orthogonality correction is performed. The rotational angular velocity is computed as:

$$\omega_i = \frac{|\varphi_i - \varphi_{i-1}|}{t_i - t_{i-1}} \quad (34)$$

Then, the time corresponding to the phase shift is:

$$\tau = \frac{|\delta|}{\omega_i} \quad (35)$$

With the ADC sampling interval, the number of samples to shift n is:

$$n = \frac{\tau}{\Delta T} \quad (36)$$

Finally, the direction of shifting is determined by the sign of the phase deviation δ , such that the two-phase inputs are adjusted to an ideal orthogonal state.

2.6. Angle Demodulation: PLL-Based Closed-Loop Tracking

2.6.1. Conventional Second-Order PLL Tracking Decoder

In software decoding, closed-loop decoding utilizes PLL techniques to accurately compute rotor angle. Beyond providing accurate angular measurements, PLL-based decoding exhibits strong immunity to interference, effectively reducing the impact of external noise on angle estimation. Therefore, it can mitigate limitations of open-loop decoding approaches, such as insufficient accuracy and sensitivity to interference. Figure 7 illustrates the conventional closed-loop tracking decoding architecture.

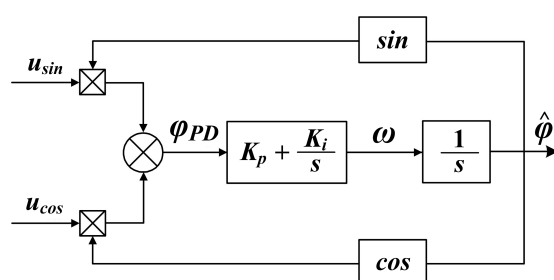


Figure 7. Block diagram of a conventional PLL-based tracking decoder.

For a PI-controlled PLL, the closed-loop transfer function is:

$$H(s) = \frac{K_p s + K_i}{s^2 + K_p s + K_i} \quad (37)$$

A PLL typically consists of a phase detector (PD), loop filter (LF), and voltage-controlled oscillator (VCO). The PD can be expressed as:

$$\varphi_{PD} = [\sin \varphi + \eta_s] \cos \hat{\varphi} - [\cos \varphi + \eta_c] \sin \hat{\varphi} = \sin(\varphi - \hat{\varphi}) + \eta \approx \varphi - \hat{\varphi} + \eta \quad (38)$$

where $\eta = \eta_s \cos \hat{\varphi} - \eta_c \sin \hat{\varphi}$ denotes system phase noise. The PD output is required to be approximately linear with respect to the phase difference for proper PLL operation.

The phase error transfer function is:

$$H_e(s) = \frac{\varphi_e(s)}{\varphi(s)} = \frac{s^2}{s^2 + K_p s + K_i} = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (39)$$

where $K_p = 2\zeta\omega_n$, $K_i = \omega_n^2$, ζ is the damping ratio and ω_n is the undamped natural frequency.

Substituting a phase step $\varphi(s) = \Delta\varphi/s$ yields Equation (40) and applying the final value theorem gives the steady-state error in Equation (41).

$$\varphi_e(s) = \frac{\Delta\varphi s}{s^2 + K_p s + K_i} \quad (40)$$

$$\lim_{t \rightarrow \infty} \varphi_e(t) = \lim_{s \rightarrow 0} s\varphi_e(s) = \lim_{s \rightarrow 0} \frac{\Delta\varphi s^2}{s^2 + K_p s + K_i} = 0 \quad (41)$$

Using $\Delta\varphi = 1$, $\zeta = 0.7$, $\omega_n = 125$, MATLAB produces the phase-step error amplitude shown in Figure 8.

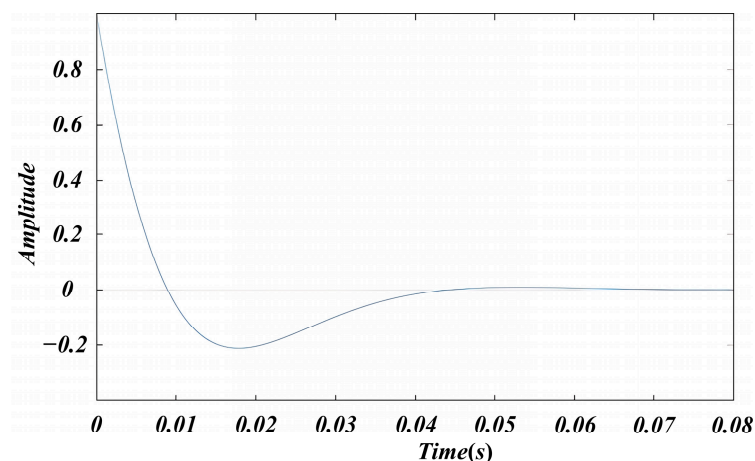


Figure 8. Phase-error response to a phase-step input (second-order PLL).

Similarly, by substituting a frequency step $\varphi(s) = \Delta\omega/s^2$ into the phase-error expression, Equation (42) is obtained, and set $\Delta\omega = 1$, $\zeta = 0.7$, $\omega_n = 125$, the corresponding MATLAB results are shown in Figure 9.

$$\lim_{s \rightarrow 0} s\varphi_e(s) = \lim_{s \rightarrow 0} \frac{s^3}{s^2 + K_p s + K_i} \frac{\Delta\omega}{s^2} = 0 \quad (42)$$

$$\lim_{s \rightarrow 0} s\varphi_e(s) = \lim_{s \rightarrow 0} \frac{s^3}{s^2 + K_p s + K_i} \frac{\Delta\omega}{s^2} = 0 \quad (43)$$

For a frequency ramp input, Equation (43) indicates that the phase error of a second-order PLL does not converge to zero; it is proportional to the ramp amplitude $\Delta\gamma$ and inversely proportional to ω_n (i.e., inversely related to the integral parameter of the PI

controller). Increasing ω_n can reduce the phase error under frequency ramp conditions. MATLAB results for ω_n settings of 125, 150, 175, and 200 are shown in Figure 10.

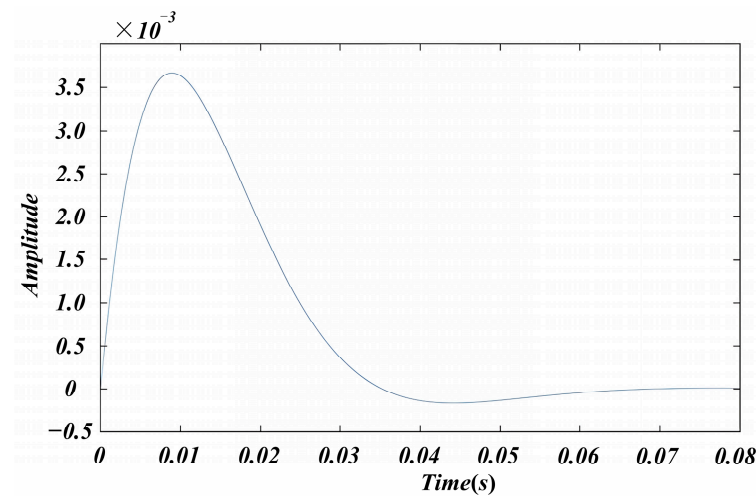


Figure 9. Phase-error response to a frequency-step input (second-order PLL).

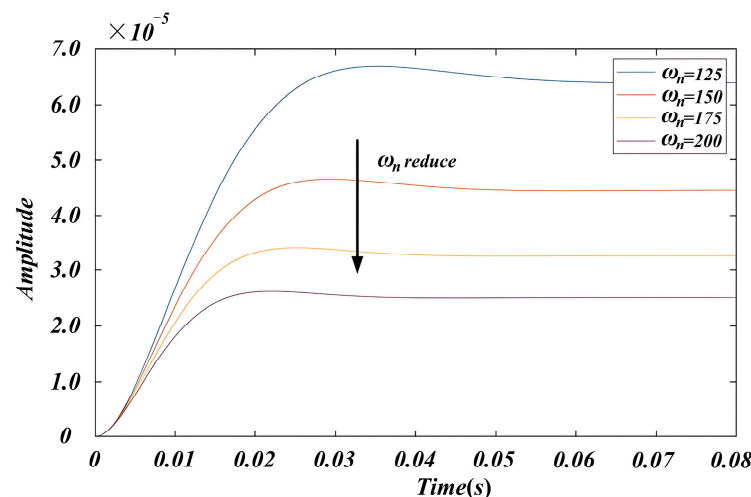


Figure 10. Phase-error amplitude under frequency-ramp input with different natural frequencies (second-order PLL).

However, the Bode plot analysis (Figure 11) shows that as ω_n increases, the gain margin decreases, making the system more prone to instability due to small gain variations.

Although increasing the integral gain can accelerate error reduction and attenuate steady-state error, an excessively large integral gain may destabilize the response, leading to larger overshoot and oscillations due to accumulated integral action. Consequently, simply increasing gains to reduce steady-state error under frequency ramp conditions is not an appropriate solution.

The above analysis indicates that when the input signal exhibits an acceleration step (i.e., a frequency ramp), DC error becomes a major contributor to phase error in second-order PLL approaches. Under a frequency ramp input, the steady-state error cannot converge to zero, implying that interference components and DC offsets directly degrade the PLL estimation accuracy. These further compromises precise rotor angle tracking and ultimately lead to an estimated rotor position that lags behind the true position.

To address these limitations, an improved third-order PLL scheme is proposed. The objective is to suppress disturbances introduced by the input signal and eliminate the impact of DC error, thereby improving overall performance and stability. This enhance-

ment is particularly important for accurate and timely rotor-state tracking under complex dynamic conditions.

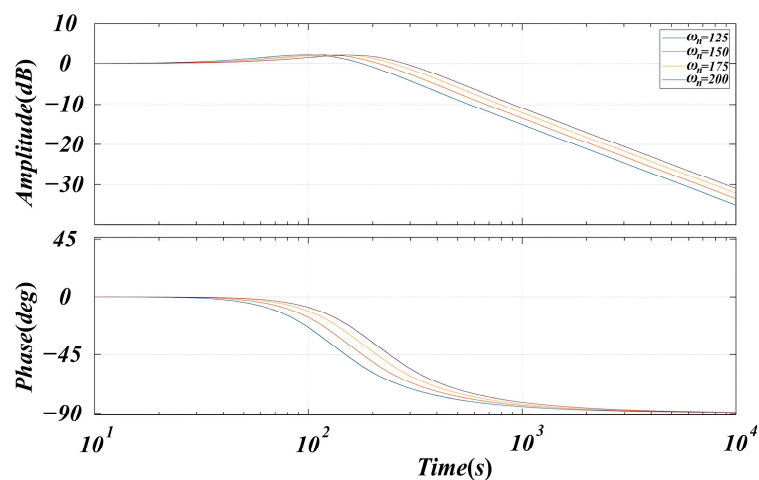


Figure 11. Bode plot illustrates stability-margin variation with PLL parameters (second-order PLL).

2.6.2. Proposed Adaptive Third-Order PLL Tracking Decoder

A third-order PLL that guarantees zero steady-state error is investigated. The proposed third-order PLL consists of a phase detector, loop filter, and VCO, as shown in Figure 12. The PD design follows that of the conventional PLL scheme. The loop filter adopts a second-order structure incorporating proportional-plus-integral factors, with transfer function:

$$F(s) = K_1 + \frac{K_2}{s} + \frac{K_3}{s^2} \quad (44)$$

where the parameters represent the gain products of the P+I factors. In general, the VCO can be modeled as an integrator. The closed-loop transfer function of the third-order PLL is:

$$H(s) = \frac{K_1 s^2 + K_2 s + K_3}{s^3 + K_1 s^2 + K_2 s + K_3} = \frac{(2+p)\zeta\omega_n s^2 + (1+2p\zeta^2)\omega_n^2 s + p\zeta\omega_n^3}{(s + p\zeta\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (45)$$

where $K_1 = (2+p)\zeta\omega_n$, $K_2 = (1+2p\zeta^2)\omega_n^2$, $K_3 = p\zeta\omega_n^3$, ω_n is the natural frequency governing the dominant frequency characteristics, ζ is the damping coefficient affecting stability and response speed, and p is the system coefficient.

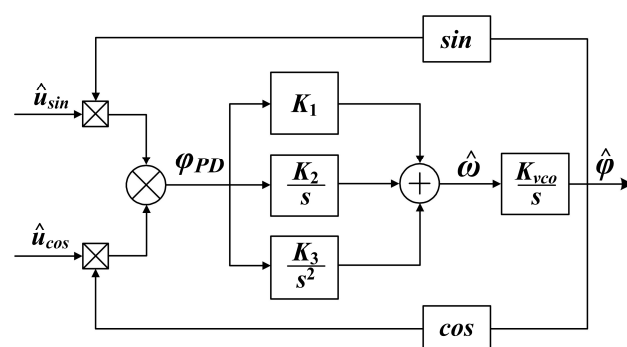


Figure 12. Block diagram of the proposed third-order PLL tracking decoder.

The phase error transfer function is:

$$H_{e_3rd}(s) = \frac{s^3}{s^3 + K_1 s^2 + K_2 s + K_3} \quad (46)$$

By substituting a phase step $\varphi(s) = \Delta\varphi/s$, there is:

$$\varphi_e(s) = \frac{\Delta\varphi s^2}{s^3 + K_1 s^2 + K_2 s + K_3} \quad (47)$$

the steady-state error derived from the final value theorem is:

$$\lim_{t \rightarrow \infty} \varphi_e(t) = \lim_{s \rightarrow 0} s \varphi_e(s) = \lim_{s \rightarrow 0} \frac{\Delta\varphi s^3}{s^3 + K_1 s^2 + K_2 s + K_3} = 0 \quad (48)$$

$$\lim_{s \rightarrow 0} s \varphi_e(s) = \lim_{s \rightarrow 0} \frac{s^4}{s^3 + K_1 s^2 + K_2 s + K_3} \frac{\Delta\omega}{s^2} = 0 \quad (49)$$

$$\lim_{s \rightarrow 0} s \varphi_e(s) = \lim_{s \rightarrow 0} \frac{s^4}{s^3 + K_1 s^2 + K_2 s + K_3} \frac{\Delta\gamma}{s^3} = 0 \quad (50)$$

Similarly, for frequency step $\varphi(s) = \Delta\omega/s^2$ and frequency ramp inputs $\varphi(s) = \Delta\gamma/s^3$, Equations (49) and (50) show that the proposed third-order PLL yields phase steady-state errors that converge to zero under all three input types (phase step, frequency step, and frequency ramp), satisfying the design objective.

To simplify parameter design, a dominant-pole approximation is adopted to reduce the third-order PLL to an equivalent second-order system. Specifically, the real part ratio of the s_3 pole to the s_1, s_2 poles is required to exceed a threshold $5p\zeta\omega_n$, ensuring s_1 and s_2 are dominant. A coefficient of p at least 5 is selected, and the practical value is set to 5.

The damping factor significantly influences dynamic performance: larger damping yields slower response but smaller overshoot; smaller damping yields faster response but larger overshoot. To achieve a well-balanced dynamic response, $\zeta = 0.707$ is selected.

Nevertheless, parameter selection involves trade-offs among acquisition time, overshoot, stability, and filtering capability. Smaller ω_n generally improves disturbance rejection at the expense of longer acquisition time, and vice versa. To ensure effective PLL operation, an adaptive algorithm is proposed to compute ω_n for the third-order PLL:

$$\omega_n(k+1) = \rho\omega_n(k) + \gamma|\varphi_{PD}(k)| \quad (51)$$

where ρ and γ are coefficients of ω_n . The initial fixed frequency $\omega_n(0)$ and the steady-state interval $[\omega_n(\min), \omega_n(\max)]$ are selected according to the experimental requirements of the system.

The adaptive update law for the natural frequency, given by

$$\omega_n(k+1) = \rho\omega_n(k) + \gamma\varphi_{PD}(k) \quad (52)$$

is designed to adjust the PLL bandwidth in response to the phase detector output. For stability, it is important that the update does not cause excessive oscillation or divergence of ω_n .

Assuming the PLL is operating near lock and the phase detector output φ_{PD} is approximately linear with respect to the phase error, the update equation forms a simple first-order discrete-time system. The parameter ρ (with $0 < \rho < 1$) acts as a smoothing factor, while γ controls the adaptation rate.

If the product γK (where K is the effective gain from phase error to φ_{PD}) is sufficiently small such that $|\rho - \gamma K| < 1$, the sequence $\omega_n(k)$ will converge exponentially to a steady-state value determined by the operating conditions. In practice, this means choosing γ small enough to ensure smooth adaptation and optionally applying bounds to ω_n to prevent instability.

To further enhance robustness, it is recommended to:

- Apply a dead-zone or low-pass filter to φ_{PD} to suppress noise-induced jitter.
- Limit the range of ω_n to physically reasonable values.
- Use a moderate ρ (e.g., 0.95–0.99) and a small γ (e.g., 10^{-4} to 10^{-2}), as validated in experiments.

With these measures, the adaptive law ensures that ω_n remains bounded and tracks the optimal value for the current operating condition, without causing instability in the PLL loop.

2.6.3. Stability and Parameter Selection of the Adaptive Natural-Frequency Update

The proposed third-order PLL uses an adaptive natural-frequency update law to vary loop bandwidth according to instantaneous tracking conditions: Equation (52), where $0 < \rho < 1$, is a smoothing factor and $\gamma > 0$ is an adaptation gain. In this subsection, we clarify (i) stability/boundedness of $\omega_n(k)$, (ii) the interaction between ω_n adaptation and PLL closed-loop stability, and (iii) a practical procedure to select $(\rho, \gamma, \omega_n(\min), \omega_n(\max))$ for embedded implementation.

(1) Boundedness and convergence of the ω_n update

Near lock, the phase detector behaves approximately linearly, i.e., $\varphi_{PD}(k) \approx K_d \cdot \varphi_e(k)$, where K_d is the effective PD gain and $\varphi_e(k)$ is the phase error. Under this local linearization, Equation (52) becomes a first-order discrete-time linear system driven by a bounded input. If $\varphi_{PD}(k)$ is bounded (which holds in practice because it is computed from bounded SIN/COS signals and is typically further limited by software), then $\omega_n(k)$ is bounded for any $0 < \rho < 1$. To guarantee global boundedness under large disturbances and measurement noise, we explicitly apply saturation:

$$\omega_n(k+1) \leftarrow \text{sat}[\omega_n(\min), \omega_n(\max)]\{\rho\omega_n(k) + \gamma\varphi_{PD}(k)\} \quad (53)$$

With Equation (53), $\omega_n(k)$ is guaranteed to remain within $[\omega_n(\min), \omega_n(\max)]$ for all k , and the adaptation cannot diverge. In addition, for a constant operating condition where the mean PD output is approximately constant, the expected value of ω_n converges to a steady-state value with an exponential rate governed by ρ .

(2) Effect of ω_n adaptation on PLL stability (bandwidth–delay trade-off)

Adapting ω_n changes the loop gains (K_1, K_2, K_3) and therefore shifts the closed-loop poles of the third-order PLL. In general, increasing ω_n increases tracking bandwidth and speeds up acquisition, but it also reduces phase margin and can amplify noise. This trade-off is well known in PLL design, and it becomes more pronounced when a moving-average filter (MAF) is used in the measurement path because the MAF introduces a fixed group delay that adds phase lag around the crossover frequency. MAF-based PLL analyses explicitly emphasize that the MAF window length (hence delay) must be co-designed with loop bandwidth to maintain adequate stability margins. (MAF-PLL design guidelines and delay effects are widely discussed in the PLL literature.)

In our preprocessing stage, we use an $N = 32$ moving-average filter. For $f_s = 100$ kHz, the linear-phase FIR group delay is

$$\tau_g = \frac{N-1}{2f_s} = \frac{15.5}{100,000} \approx 155 \mu\text{s} \quad (54)$$

This delay contributes an additional phase lag of approximately $\Delta\varphi(\omega) \approx -\omega\tau_g$. Therefore, to preserve stability while allowing fast tracking, we constrain ω_n within a

safe upper bound such that $\omega_{BW}\tau_g$ remains small (a practical engineering guideline is $\omega_{BW}\tau_g \leq 0.1$). Taking $\omega_{BW} \approx \omega_n$ for dominant-pole designs yields:

$$\omega_n(max) \leq 0.2/\tau_g \approx 0.2/(155 \times 10^{-6}) \approx 1.29 \times 10^3 \text{ rad/s } (\approx 205 \text{ Hz}) \quad (55)$$

This bound prevents excessive phase-margin erosion caused by the MAF delay, while still allowing sufficiently fast transient response under commutation and acceleration. The lower bound $\omega_n(min)$ is set to ensure adequate disturbance rejection and noise filtering in steady operation.

(3) Practical selection of $(\rho, \gamma, \omega_n(min), \omega_n(max))$

A practical selection procedure is as follows:

Step 1 (choose $\omega_n(max)$) from delay and stability: Use Equation (55) to select $\omega_n(max)$ so that the added delay-induced phase lag around crossover does not compromise stability. In our system with $\tau_g \approx 155 \mu\text{s}$, $\omega_n(max)$ on the order of 10^3 rad/s is a conservative upper limit.

Step 2 (choose $\omega_n(min)$ from steady-state noise and ripple): $\omega_n(min)$ should be small enough to attenuate measurement noise and harmonics that leak through the front-end, but not so small that it causes sluggish response. In practice, $\omega_n(min)$ is selected by evaluating steady-state angle jitter (or phase-error variance) under constant speed and choosing the smallest ω_n that still meets dynamic requirements.

Step 3 (choose ρ from desired adaptation time constant): The parameter ρ sets the “memory” of $\omega_n(k)$. For sampling period $T_s = \frac{1}{f_s} = 10 \mu\text{s}$, the adaptation time constant can be approximated as:

$$T_{adapt} \approx T_s/(1 - \rho) \quad (56)$$

For example, $\rho = 0.99$ gives $T_{adapt} = 1 \text{ ms}$, while $\rho = 0.995$ gives $T_{adapt} = 2 \text{ ms}$. Larger ρ reduces ω_n jitter and makes the adaptation smoother, while smaller ρ accelerates ω_n changes but may amplify noise-induced fluctuations.

Step 4 (choose γ from adaptation speed and PD scaling): The gain γ sets how strongly φ_{PD} affects ω_n . To avoid ω_n oscillation due to noise, γ should satisfy a small-gain condition under the near-lock linearization. If $\varphi_{PD} \approx K_d \varphi_e$ and the effective gain from ω_n perturbation to φ_{PD} is bounded, then choosing γ sufficiently small ensures the ω_n update remains well damped. In implementation, γ is tuned by applying a step change in speed (or commutation event) and increasing γ until the ω_n transient becomes responsive but not oscillatory; saturation Equation (53) and optional dead zone on φ_{PD} can further suppress noise-driven jitter.

(4) Implementation notes

To improve robustness in fixed-point embedded code, we additionally recommend:

- (i) apply a dead zone or low-pass filter to $\varphi_{PD}(k)$ before Equation (53) to reduce noise-driven ω_n jitter;
- (ii) enforce Equation (53) with $\omega_n(min)/\omega_n(max)$ to guarantee bounded ω_n ;
- (iii) update ω_n at a slower rate than the ADC sampling rate if CPU budget is tight (e.g., update every M samples), which further smooths ω_n without affecting the main PLL numerical integration.

Overall, the adaptive law Equations (52) and (53) can be interpreted as a bounded, first-order bandwidth scheduler; ω_n is temporarily increased during large phase errors to speed up acquisition and then reduced in steady state to improve noise filtering and preserve stability margins. This mechanism is consistent with prior MAF-based PLL design guidance and the general bandwidth–delay trade-off in practical PLL systems.

2.6.4. Baseline Demodulators for Comparison

To enable a fair and representative evaluation beyond the conventional second-order PLL (Section 2.6.1), we introduce two additional baselines that are commonly used in sinusoidal encoder decoding.

Baseline 1: Non-adaptive third-order PLL (fixed ω_n).

This baseline uses the same third-order PLL loop structure as Figure 12 and the same dominant-pole design (p and ς) as the proposed method but removes the adaptive natural-frequency update law in Equations (51) and (52). Instead, the natural frequency ω_n is fixed to a constant value ω_n , fix throughout the experiment. The loop-filter coefficients K_1 – K_3 are computed using the same relations as in Equation (45). This baseline isolates the benefit of the third-order loop order from the benefit of bandwidth adaptivity.

Baseline 2: Compensation + arctangent (open-loop).

This baseline applies to the identical front-end compensation pipeline described in Section 2.5 (fixed-window moving average, amplitude normalization, and phase orthogonality identification with sample-shift correction). After compensation, the angle is estimated by open-loop arctangent demodulation as $\hat{\theta} = \text{atan2}(\hat{u}\sin, \hat{u}\cos)$. This baseline quantifies how much accuracy improvement can be achieved by compensation alone without PLL-based closed-loop tracking.

3. Experimental Setup

3.1. PMSM Bench, Sensors, Magnet, MCU, and Sampling Configuration

3.1.1. Mechanical and Drive Bench

Experiments were conducted on a high-speed PMSM bench designed to expose the encoder to steady rotation, direction commutation, and frequency ramp inputs. The bench comprises a 200 W permanent-magnet synchronous motor (Haitai, Changzhou, China), driven by a custom STM32-based controller board (STMicroelectronics, Geneva, Switzerland). The motor's electrical parameters are listed in Table 2. A rigid fixture holds the motor, encoder PCB, and magnet assembly to minimize mechanical runout while allowing controlled misalignment tests.

Table 2. Parameters of the PMSM used in the experiments.

Parameters	Value	Unit
Rated power	200	W
Pole pairs	2	-
Rated voltage	36	V
Rated current	6.5	A
Line resistance	0.44	Ω
Line inductance	0.45	mH
Back-EMF constant	6.3	V/kRPM
Tamagawa encoder precision	24	Bit

3.1.2. Magnet and Sensors

A single-pole-pair radially magnetized disk magnet (diameter 10 mm, thickness 3 mm) is mounted coaxially to the rotor shaft. The sensing head uses a biaxial full-bridge TMR chip (TMR3081, Dowaytech, Suzhou, China) arranged to generate two differential channels nominally in quadrature (X/Y axes). To evaluate robustness under nonideal conditions, the PCB includes shims to intentionally offset the TMR chip from the magnet center; offset magnitude and orientation were varied in repeat runs to induce amplitude mismatch,

phase non-orthogonality, and harmonic content representative of industrial tolerances. The mechanical stack and DUT photographs are shown in Figure 13.

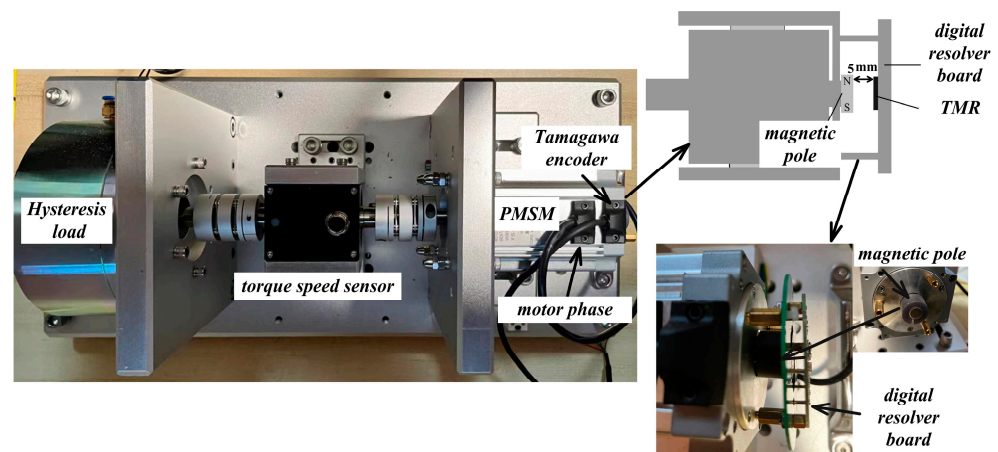


Figure 13. Experimental testbench and mechanical structure of the TMR magnetic encoder prototype.

3.1.3. Controller, Acquisition, and Fixed-Point Preprocessing

The STM32 microcontroller samples the two differential channels via on-chip ADCs. Unless otherwise noted, the following preprocessing pipeline is executed in real time before demodulation:

- Set sample frequency to 100 kHz. Fixed-window moving average with a window of 32 samples (constant-time update using a circular buffer). The 32-point choice balances noise suppression with latency, and it enables efficient fixed-point division by a right-shift of five bits.
- Linear amplitude normalization mapping each channel to a common range 0–2000 counts to suppress amplitude mismatch while preserving waveform shape.
- Phase orthogonality identification and sample-shift compensation: the instantaneous phase difference is estimated from one-period correlations; the required sample shift is applied according to the sign of phase deviation to approach 90° quadrature (see Section 2.5.3 for the derivation).
- Adaptive third-order PLL tracking for angle demodulation, using the loop structure in Section 2.6.2 and the adaptive natural-frequency update in Equation (51).

To verify front-end signal integrity and the effect of compensation, raw and processed channel outputs are exposed on test pins and captured by a digital oscilloscope. For Lissajous inspection (orthogonality check), channel-A and channel-B are fed to X/Y inputs; a circular trace indicates ideal quadrature, while ellipses indicate phase deviation. For quantitative post-analysis, oscilloscope traces are exported and parsed offline to reproduce figures in Section 4.4 (raw vs. compensated waveforms) and the error statistics in Section 4.5.

3.2. Reference Encoder and Metrics

3.2.1. Reference and Ground Truth

For ground-truth angle, the bench is equipped with a Tamagawa 24-bit absolute encoder mounted on the same shaft. Its digital angle output is time-aligned with the TMR-based estimate using the controller's timestamp, ensuring consistent sampling instants for error computation. The Tamagawa encoder's resolution provides a sufficiently fine reference to characterize sub-degree differences across the tested speed range.

3.2.2. Primary Evaluation Metrics

All metrics are computed over clearly annotated segments under identical operating conditions for the baseline second-order PLL decoder and the proposed adaptive third-order PLL decoder. Angle is reported in degrees unless stated otherwise.

For each scenario, we mark a steady window (excluding spin-up and reversal intervals) and compute DC Bias/MAE/RMSE/Max Error. For commutation and ramp tests, we isolate a transient window to compute overshoot. All statistics are reported with consistent units (degrees for angle, percent for overshoot). Where applicable, raw traces and processing scripts are archived to enable reproducibility.

3.2.3. Compared Algorithms and Fair-Comparison Protocol

We compare the following decoders under identical operating segments and the same reference alignment:

- (A0) Conventional second-order PLL demodulator (Section 2.6.1).
- (A1) (Proposed compensation pipeline (Section 2.5) + open-loop arctangent demodulation (Baseline B2).
- (A2) Proposed compensation pipeline (Section 2.5) + non-adaptive third-order PLL with fixed ω_n (Baseline B1).
- (A3) Proposed compensation pipeline (Section 2.5) + adaptive third-order PLL (Section 2.6.2, proposed).

For fairness, all compared methods use the same sampled raw data and the same time alignment to the 24-bit Tamagawa encoder reference. All metrics (DC bias, MAE, RMSE, maximum absolute error, and transient overshoot) are computed over identical annotated windows for each operating scenario (steady segments exclude spin-up and reversal; transient segments isolate commutation/ramp intervals).

Figure 14 summarizes a physical–computational integration pipeline for the TMR magnetic encoder, explicitly linking dominant physical non-idealities (amplitude mismatch, non-orthogonality, harmonics/noise, and delay) to lightweight identification/compensation modules and to the downstream demodulation baselines (A0–A3) for fair evaluation.

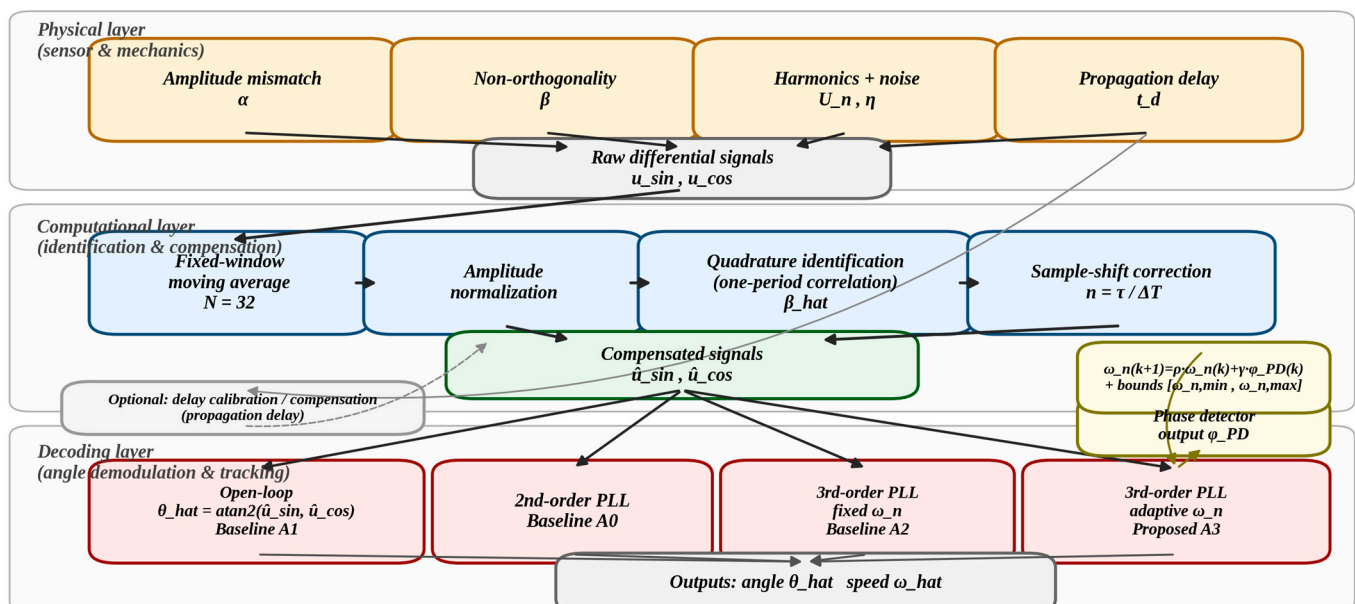


Figure 14. Physical–computational integration framework that maps TMR encoder non-idealities to compensation/identification steps and to the compared demodulators (A0–A3).

4. Results

4.1. Amplitude Normalization Performance

As described in Section 2.5, amplitude normalization is used to address unequal amplitudes of the two differential outputs. In this experiment, test points reserved in the hardware design are used to output the raw sensor readings via pins. The waveforms are captured by an oscilloscope, exported in CSV format, and plotted, as shown in Figure 15. It is evident that the two-phase differential signals have unequal amplitudes.

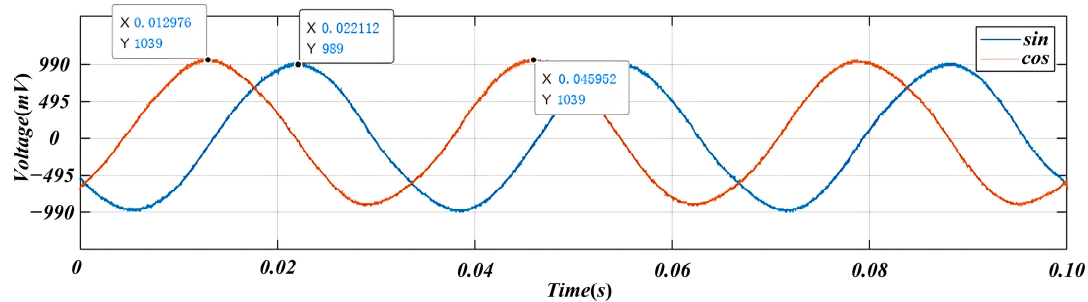


Figure 15. Measured raw differential waveforms before amplitude normalization.

After applying amplitude normalization, the results in Figure 16 show that both signals are mapped into the range 0–2000 with equalized amplitudes, thereby mitigating errors caused by amplitude mismatch.

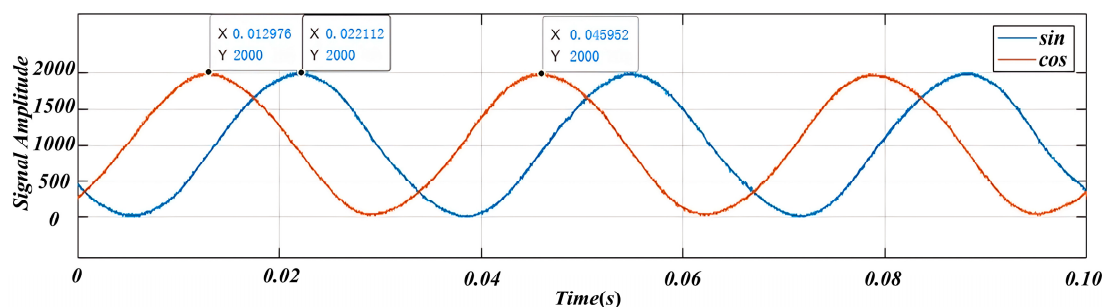


Figure 16. Measured waveforms after amplitude normalization.

4.2. Phase Non-Orthogonality Compensation via Lissajous Patterns

A Lissajous figure is a method for visualizing the composite pattern formed by two sinusoidal vibrations along orthogonal axes on an oscilloscope. By examining the resulting pattern, one can infer properties such as frequency ratio and phase difference. When two sine waves are orthogonal, they have a 90° phase difference; in this case, the Lissajous figure becomes a circle. Practically, by feeding one channel into the oscilloscope X input and the other into the Y input, a circular trace indicates identical frequency and 90° phase difference, i.e., orthogonality.

Figure 17 compares the Lissajous figures before and after compensation. Before compensation, the trace is elliptical, indicating non-orthogonality. After compensation, the trace becomes closer to an ideal circle, demonstrating that the phase compensation is effective.

4.3. Combined Compensation During Commutation

The combined error-compensation approach in this design proceeds as follows: (i) apply a fixed-window moving average filter to the two differential voltage signals to smooth data and suppress noise; (ii) apply amplitude normalization to equalize the amplitudes; (iii) assess phase orthogonality and apply compensation if necessary.

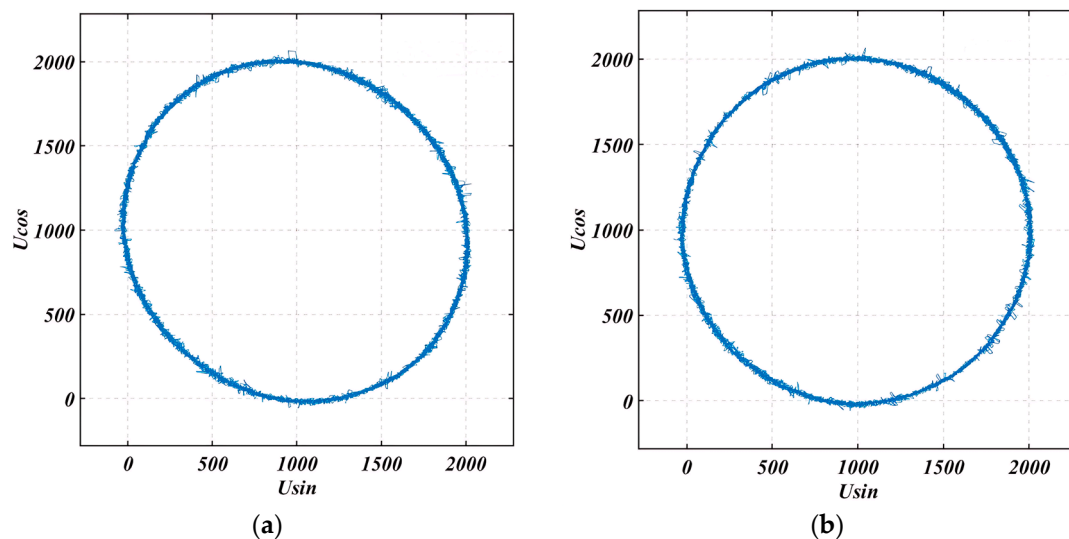


Figure 17. Lissajous figures before and after phase orthogonality compensation. (a) Before compensation; (b) After compensation.

Figure 18 shows the raw waveform of the angle sensor output when the motor rotation direction changes. Visible clutter/noise is present, which can substantially affect subsequent angle computation. In addition, amplitude inconsistency between the two phases is observed, likely due to non-level soldering of the sensor or mechanical installation tolerances of the PCB.

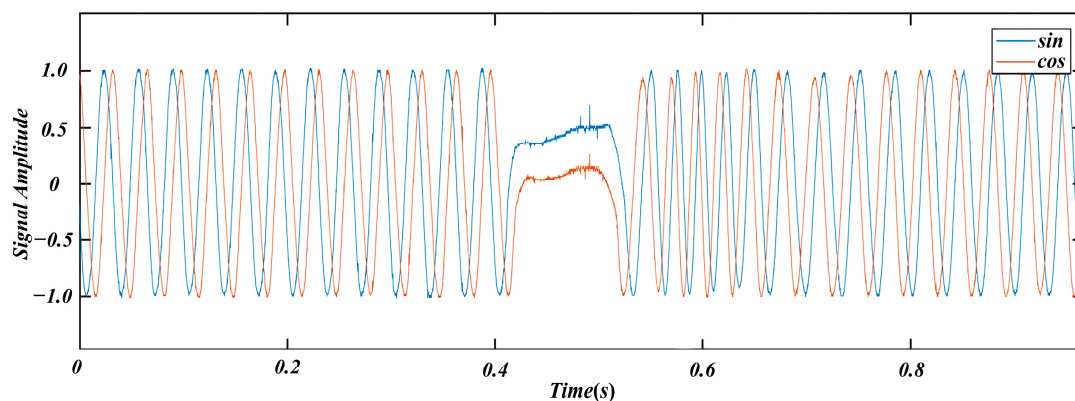


Figure 18. Raw waveforms during commutation before combined compensation.

After applying the combined compensation, the waveform during commutation is improved, as shown in Figure 19.

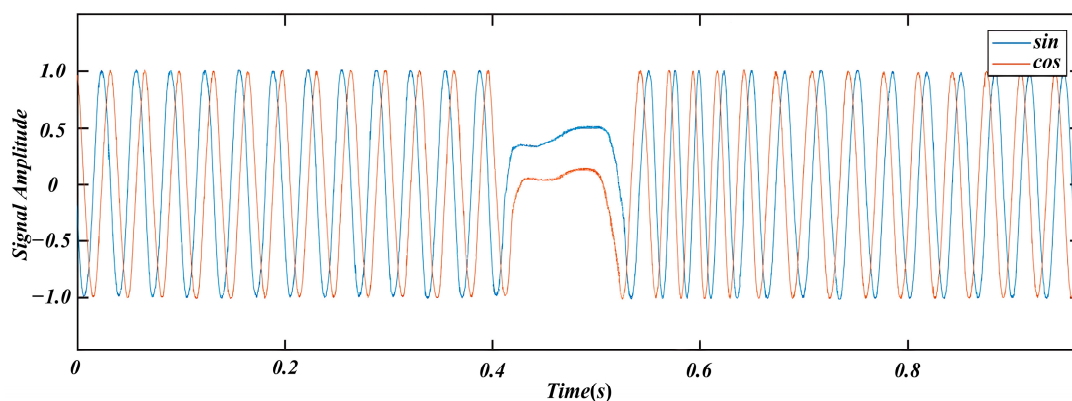


Figure 19. Waveforms during commutation after combined compensation.

4.4. Tracking Demodulation Accuracy: Conventional vs. Proposed PLL

Figure 20 shows the steady-state error when the conventional closed-loop tracking decoding algorithm is used. A DC-error of approximately 0.015 rad is observed, with the error magnitude around $0\sim 1.7^\circ$, consistent with the theoretical analysis.

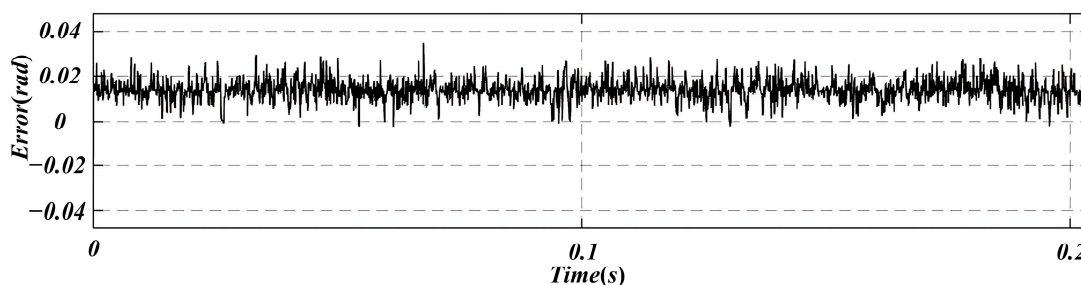


Figure 20. Steady-state tracking error with the conventional PLL-based decoder.

Figure 21 shows the steady-state error using the improved closed-loop tracking decoding algorithm. When the motor speed is stable, the DC error is eliminated, and the error is approximately within $\pm 0.3^\circ$.

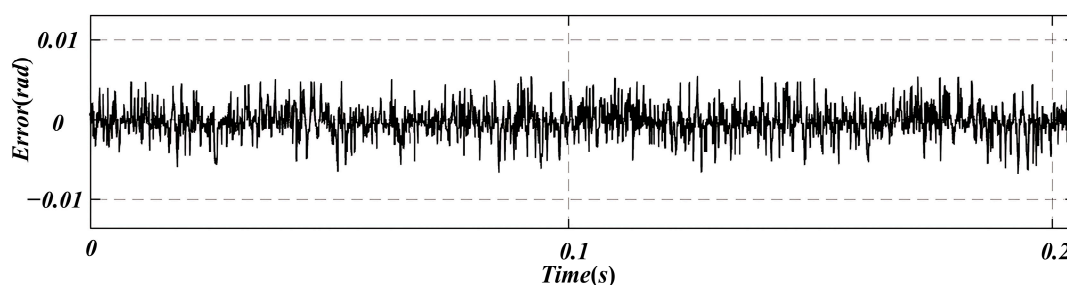


Figure 21. Steady-state tracking error with the proposed decoder.

Figures 22 and 23 illustrate the phase error during commutation (i.e., rotation direction change) using the conventional and improved decoding algorithms, respectively. The results indicate that the proposed third-order PLL-based approach removes more DC error than the second-order PLL and remains effective during commutation. Figure 24 shows the final output angle after processing with the improved decoding algorithm.

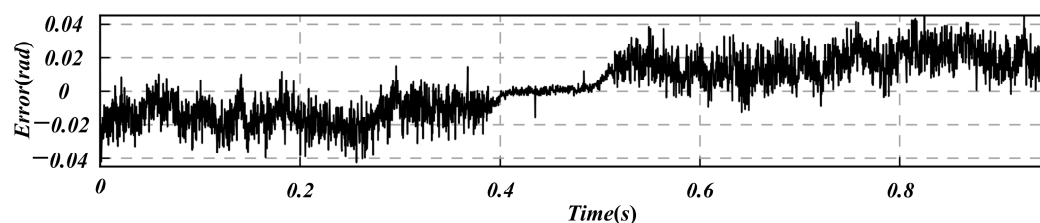


Figure 22. Phase error during commutation using conventional 2nd-order PLL (A0).

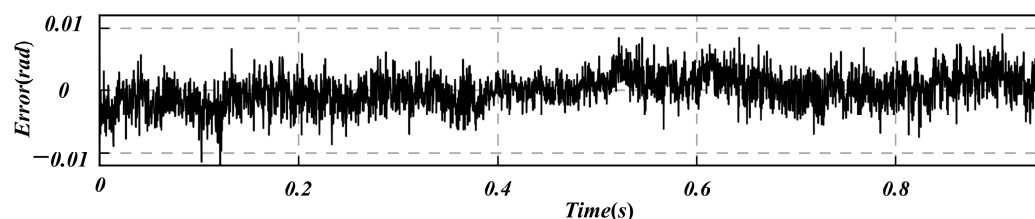


Figure 23. Phase error during commutation using adaptive 3rd-order PLL (proposed) (A3).

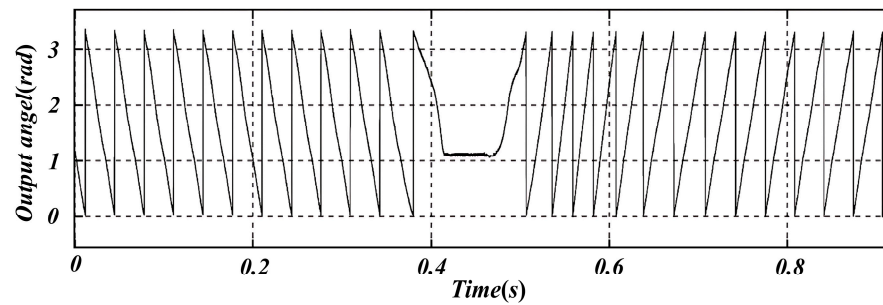


Figure 24. Final reconstructed output angle using the proposed decoder.

4.5. Extended Baselines: Non-Adaptive Third-Order PLL and Compensated Arctangent

To address the concern that a second-order PLL alone is not a sufficiently representative baseline, we evaluate two additional decoders defined in Section 2.6.4: (i) a non-adaptive third-order PLL with fixed ω_n , and (ii) a compensated open-loop arctangent demodulator. All methods share the same front-end compensation chain (Section 2.5) when applicable and are evaluated using the same reference alignment and metric windows (Section 3.2).

Figures 25 and 26 illustrate the two added baselines. Figure 27 (and Table 3) summarizes the quantitative comparison across A0–A3. The results show that compensation alone (arctangent baseline) reduces systematic errors caused by amplitude mismatch and non-orthogonality but closed-loop tracking remains advantageous during commutation and ramp-like dynamics. Furthermore, the fixed- ω_n third-order PLL improves over the second-order PLL by reducing DC bias under dynamic inputs, while the adaptive third-order PLL achieves the best overall trade-off by adjusting bandwidth according to the phase detector output and maintaining bounded ω_n within predefined limits.

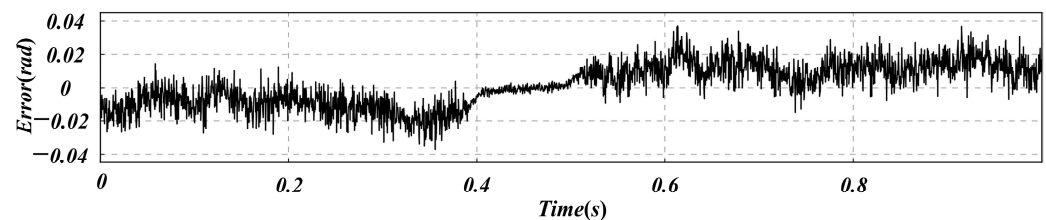


Figure 25. Phase error during commutation using atan2 (open loop) (A1).

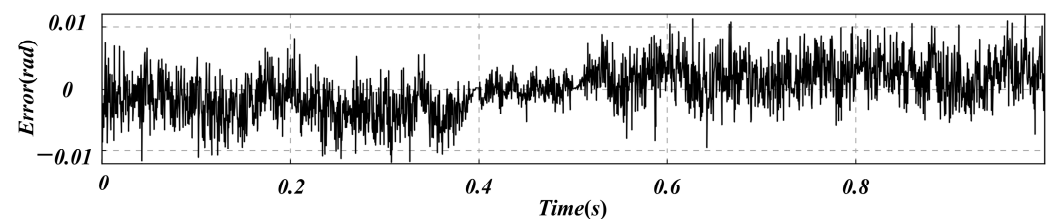


Figure 26. Phase error during commutation using fixed- ω_n 3rd-order PLL (non-adaptive) (A2).

Table 3. Quantitative error metrics and transient overshoot comparison for demodulation baselines A0–A3.

Case	DC Bias (Deg)	MAE (Deg)	RMSE (Deg)	Max Error (Deg)	Overshoot (%) (Commutation)	Overshoot (%) (Ramp)
A0	0.0275	0.1862	0.2120	0.4562	32.9214	203.3824
A1	0.0144	0.1382	0.1603	0.3816	9.0296	180.5984
A2	0.0153	0.4683	0.5817	1.7646	60.3912	1906.7719
A3	−0.0266	0.2996	0.3772	1.3904	3334.9695	701.9422

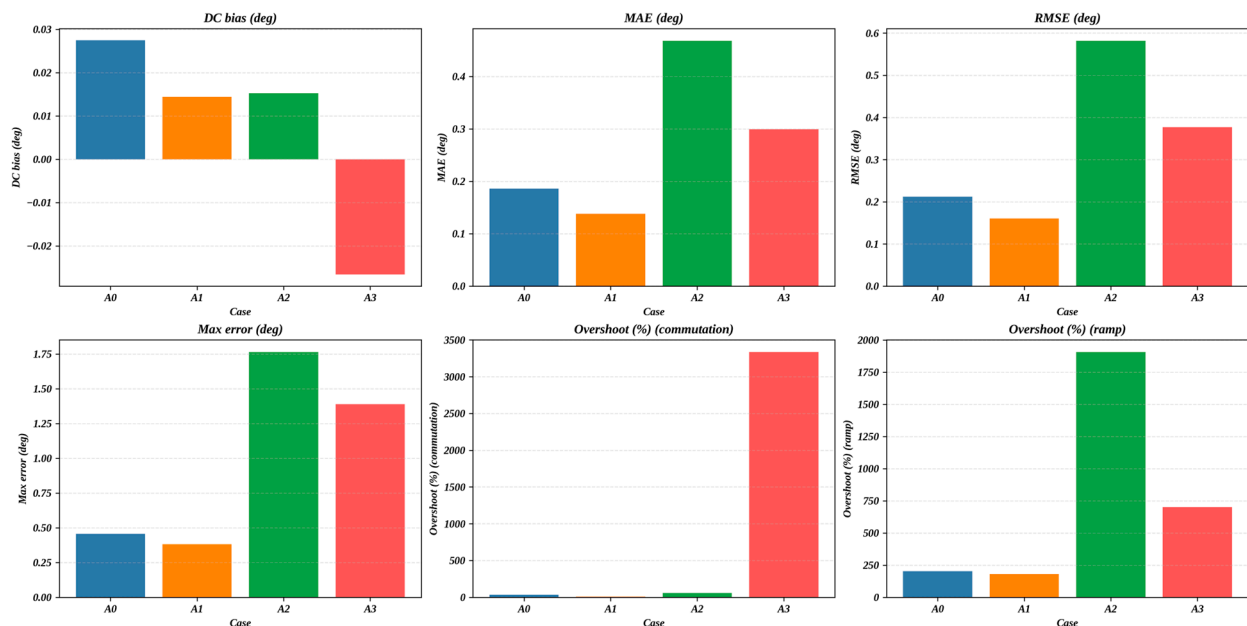


Figure 27. Visual comparison of six evaluation metrics (DC bias, MAE, RMSE, max error, commutation overshoot, ramp overshoot) across A0–A3.

Quantitative comparison of A0–A3 (steady-state and transient metrics). To ensure a fair comparison, four demodulation configurations (A0–A3) are evaluated under identical data acquisition and reference alignment conditions: A0 is the conventional second-order PLL demodulator, A1 is the proposed front-end compensation pipeline followed by open-loop arctangent demodulation, A2 is the proposed compensation pipeline with a non-adaptive third-order PLL using fixed ω_n , and A3 is the proposed compensation pipeline with the adaptive third-order PLL. All metrics in Table 3 are computed over the same annotated windows, where DC bias, MAE, RMSE, and maximum absolute error characterize steady-state accuracy, while overshoot is computed over the designated transient windows for commutation and ramp dynamics and summarized in Figure 27.

Steady-state accuracy (DC bias/MAE/RMSE/Max error). As reported in Table 3 and visualized in Figure 27, the four methods exhibit small DC offsets overall, but differ noticeably in average-error and peak-error statistics. In this dataset, A1 achieves the lowest MAE and RMSE, indicating that the compensation chain can substantially suppress the dominant systematic distortions (e.g., amplitude mismatch and non-orthogonality) even without closed-loop tracking, thereby reducing the average error magnitude and the quadratic error energy. By contrast, the third-order PLL baselines (A2/A3) do not necessarily minimize the global steady-state MAE/RMSE in this specific evaluation window, suggesting that increasing loop order (and introducing bandwidth scheduling) primarily targets dynamic tracking robustness rather than guaranteeing a minimum in aggregated steady-segment error energy.

Transient behavior (overshoot under commutation and ramp windows). Overshoot is computed using $|\text{steady} - \text{initial}|$ in the prescribed commutation and ramp windows, and the results are summarized in Table 3 and Figure 27. It is important to note that is highly sensitive to the denominator: when a transient window exhibits only a small net change between “initial” and “steady,” the overshoot percentage can become numerically large even if the absolute error remains within a moderate range. Therefore, the overshoot percentages should be interpreted jointly with Max error (deg) (and, if needed, with peak-to-peak error in degrees) to avoid overemphasizing percentage inflation due to small step magnitude within the selected window. Overall, Figure 27 highlights the trade-off among A0–A3: compensation-only decoding can excel in reducing steady-segment error statistics,

whereas PLL-based closed-loop designs are principally motivated by maintaining stable tracking under commutation and acceleration-like conditions, where loop dynamics and bandwidth selection become decisive.

5. Discussion

5.1. What Is New Compared with Prior TMR Encoder Work and Third-Order PLL Decoders

Many studies address TMR encoder errors through calibration or compensation. Others improve tracking by using higher-order PLL/DPLL structures. The contribution here is not a single “new filter” or “a higher-order PLL” in isolation. The novelty is the end-to-end integration in a concrete embedded prototype. The compensation steps are deliberately lightweight and fixed-point friendly. The decoder is a third-order PLL with adaptive natural frequency to balance acquisition and steady jitter under our sampling and delay constraints.

A second point is the evaluation protocol. We run the second-order PLL baseline and the proposed method under the same acquisition, the same reference alignment, and the same metric windows. We also include two additional baselines (compensated arctangent and fixed- ω_n third-order PLL). This helps separate the effect of compensation from the effect of PLL order and the effect of bandwidth adaptation. In our dataset, compensation alone reduces average error metrics, while closed-loop tracking is more beneficial during commutation and ramp-like dynamics.

5.2. Interpretation of the Experimental Outcomes

The measurements support two practical observations for this prototype. First, amplitude mismatch produces a repeatable second-harmonic error component. Simple normalization reduces this component effectively. Second, non-orthogonality is more damaging than amplitude mismatch in our setup. The correlation-based phase identification plus sample shifting improves quadrature, which is also consistent with the Lissajous plots becoming closer to a circle.

On the tracking side, the experiments match the control-theoretic expectation. The second-order PLL can show residual bias under ramp-like inputs. Increasing loop bandwidth can reduce the bias, but it may reduce stability margins when a moving-average filter introduces delay. The third-order PLL structure removes the steady-state error to ramp inputs in theory. In the bench results, the adaptive third-order PLL reduces the DC bias seen in the second-order PLL during steady segments and improves robustness during commutation events.

5.3. Limits of This Study (Scope and Non-Generality)

This work is limited to a specific prototype configuration. The sensor is a TMR3081 with one magnet geometry and one mechanical assembly. The sampling frequency and moving-average window (100 kHz and $N = 32$) are fixed in this implementation. The adaptive ω_n bounds are selected based on our delay and stability considerations. Different sensors, magnets, air gaps, misalignment patterns, or MCU resources may lead to different trade-offs and different optimal parameters.

The error statistics also depend on the selected windows and metrics. In particular, overshoot percentages can be inflated when the net step in a transient window is small. Therefore, overshoot should be interpreted together with absolute peak error in degrees. This paper does not claim universal superiority across all encoders or all operating profiles. It reports what is observed on this bench and under these settings.

5.4. Practical Implications for Follow-Up Work

Several improvements are possible within the same framework. Harmonic-rich cases could benefit from explicit harmonic estimation or selective suppression. Propagation-delay

compensation could be strengthened by a calibration routine or online identification. The ω_n adaptation law could also incorporate a dead-zone or filtered phase detector output to reduce jitter at very low speed.

6. Conclusions

This paper studied a TMR SIN/COS magnetic encoder prototype on a PMSM bench. We focused on combined effects from front-end nonidealities and PLL tracking dynamics.

1. We analyzed dominant error sources in the prototype signals, including amplitude mismatch, quadrature non-orthogonality, and harmonic/noise disturbances.
2. We implemented a lightweight compensation chain suitable for embedded execution. It uses moving-average denoising, amplitude normalization, and correlation-based quadrature identification with sample-shift correction.
3. We implemented an adaptive natural-frequency third-order PLL decoder and compared it with conventional second-order PLL using the same data alignment to a 24-bit reference encoder.

In the reported experiments, the proposed approach removes the steady DC bias observed in the second-order PLL and reduces commutation-induced phase deviation. The steady-state angular error is within approximately $\pm 0.3^\circ$ in our evaluated windows. These findings are limited to this prototype and parameter set. They primarily show that, for this implementation, co-designing simple compensation with a higher-order adaptive PLL can improve practical tracking behavior under steady and commutation conditions.

Author Contributions: Conceptualization, Y.X. and J.C.; methodology, J.C.; software, L.L. and Y.X.; validation, L.L.; formal analysis, J.C.; writing—original draft preparation, Y.X.; writing—review and editing, J.C. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 51977101; Zhenjiang City Industrial Prospective and Common Key Technology Research and Development Project, grant number GY2023003.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We sincerely thank the reviewers for taking the time to review this paper amidst their busy schedules.

Conflicts of Interest: Author Li Liu was employed by the company Suzhou Veichi Electric Co., Ltd. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhou, B.; Huang, C. Magnetic Poles Position Detection of Permanent Magnet Linear Synchronous Motor Using Four Linear Hall Effect Sensors. *Actuators* **2023**, *12*, 269. [\[CrossRef\]](#)
2. Demirci, E. Magnetic and Magnetotransport Properties of Memory Sensors Based on Anisotropic Magnetoresistance. *J. Supercond. Nov. Magn.* **2020**, *33*, 3835–3840. [\[CrossRef\]](#)
3. Lumetti, S.; Malagò, P.; Stürmer, P.-A.; Relvão, F.F.; Ortner, M. AMR Sensor Array Design for the Realization of a 3D Magnetic Tactile Sensor. *Proceedings* **2024**, *97*, 198. [\[CrossRef\]](#)
4. Ger, T.-R.; Wu, P.-S.; Wang, W.-J.; Chen, C.-A.; Abu, P.A.R.; Chen, S.-L. Development of a Microfluidic Chip System with Giant Magnetoresistance Sensor for High-Sensitivity Detection of Magnetic Nanoparticles in Biomedical Applications. *Biosensors* **2023**, *13*, 807. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Tan, X.; Li, Z.; Gao, X.; Wu, H.; Huang, D.; Liu, W. Structures of Giant Magnetoresistance Sensors for Magnetic Bead Detection. *IEEE Sens. J.* **2024**, *24*, 36587–36593. [\[CrossRef\]](#)
6. Mendonsa, R.; Liang, S.; Tonini, D.; Wang, J.-P. Micromagnetic Simulations of Concentration Detection Capability of Large-Area and Low-Aspect-Ratio GMR Sensors in Biomedical Diagnostics. *IEEE Trans. Magn.* **2025**, *61*, 5100204. [\[CrossRef\]](#)

7. Yan, S.; Zhou, Z.; Yang, Y.; Leng, Q.; Zhao, W. Developments and Applications of Tunneling Magnetoresistance Sensors. *Tsinghua Sci. Technol.* **2022**, *27*, 443–454. [\[CrossRef\]](#)
8. Khan, A.U.; Alzahrani, A. A Novel TMR Cantilever-Based Bi-Directional Flow Sensor for Agricultural and Domestic Applications. *Sensors* **2025**, *25*, 5915. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Tsukada, K.; Hayashi, M.; Nakamura, Y.; Sakai, K.; Kiwa, T. Small Eddy Current Testing Sensor Probe Using a Tunneling Magnetoresistance Sensor to Detect Cracks in Steel Structures. *IEEE Trans. Magn.* **2018**, *54*, 1–5. [\[CrossRef\]](#)
10. Liu, N.; Du, B.; Mei, J. Noncontact Current Measurement Based on 3-D TMR Sensor Array. *IEEE Sens. J.* **2025**, *25*, 33519–33527. [\[CrossRef\]](#)
11. Qi, Z.; Wei, P.; Liu, C.; Huang, H.; Xu, H.; Li, X. A Novel Tunneling Magnetoresistive Current Sensor Based on Reference Magnetic Field Source. *IEEE Sens. Lett.* **2024**, *8*, 5500304. [\[CrossRef\]](#)
12. Zhao, G.; Lei, B.; Ye, G.; Ban, Y.; Li, X.; Liu, H.; Liu, H. Improved Eccentricity Self-Detection Method Based on Least Square Algorithm for Polar Coordinate Encoder. *IEEE Sens. J.* **2021**, *21*, 26902–26911. [\[CrossRef\]](#)
13. Gong, J.; Liu, D.; Luo, J. A Positioning System Design Based on Tunnel Magnetoresistance Sensors for Rapid Zoom Optical Lens. *Sensors* **2025**, *25*, 1820. [\[CrossRef\]](#)
14. Ye, S.; Wang, C.; Pang, R.; Qiu, B. TMR Nonlinear Compensation Method Based on Odd Harmonic Components. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 4509710. [\[CrossRef\]](#)
15. Lee, Y.; Kim, S.H.; Lee, S.-H.; Chung, C.C. Encoder Calibration Method for High Precision Servo Systems with a Sinusoidal Encoder. *IEEE Trans. Ind. Electron.* **2022**, *69*, 752–762. [\[CrossRef\]](#)
16. Niu, Q.; Li, R.; Lin, S.; Li, S.; Lin, S.; Niu, J. Reflective Grating Linear Encoder Based on Moiré Signals Reconstruction and Arctangent Subdivision Method. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 1016910. [\[CrossRef\]](#)
17. Zhao, G.; Ban, Y.; Zhang, Z.; Shi, Y.; Liu, H. Phase Demodulation Strategy Based on Kalman Filter for Sinusoidal Encoders. *IEEE Sens. J.* **2023**, *23*, 10625–10632. [\[CrossRef\]](#)
18. Li, Q.; Xiong, S.; Wang, S.; Dong, X.; Zhang, H. Design and Simulation of Magnetic Shielding Structure Based on Closed-Loop TMR Current Sensor. *Micromachines* **2025**, *16*, 272. [\[CrossRef\]](#)
19. Qian, S.; Guo, J.; Huang, H.; Chen, C.; Wang, H.; Li, Y. Measurement of Small-Magnitude Direct Current Mixed with Alternating Current by Tunneling Magnetoresistive Sensor. *IEEE Sens. Lett.* **2022**, *6*, 5501004. [\[CrossRef\]](#)
20. Janiszewski, D. An FPGA-Based Trigonometric Kalman Filter Approach for Improving the Measurement Quality of a Multi-Head Rotational Encoder. *Energies* **2024**, *17*, 6122. [\[CrossRef\]](#)
21. Smadi, I.A.; Kreashan, H.A.; Atawi, I.E. Enhancing the Filtering Capability and the Dynamic Performance of a Third-Order Phase-Locked Loop under Distorted Grid Conditions. *Energies* **2023**, *16*, 1472. [\[CrossRef\]](#)
22. Silva, J.; Caetano, D.M.; Rabuske, T.; Cardoso, S.; Piedade, M.; Fernandes, J.R. Integrated Circuit for Magnetic Encoder Sensing in TMR-Based Industrial Positioning System. *IEEE Trans. Ind. Electron.* **2024**, *71*, 9904–9913. [\[CrossRef\]](#)
23. Nguyen, H.X.; Tran, T.N.-C.; Park, J.W.; Jeon, J.W. An Adaptive Linear-Neuron-Based Third-Order PLL to Improve the Accuracy of Absolute Magnetic Encoders. *IEEE Trans. Ind. Electron.* **2019**, *66*, 4639–4649. [\[CrossRef\]](#)
24. Wang, Y.; Bao, X.; Hua, W.; Liu, K.; Wang, P.; Hu, M. Implementation of Embedded Magnetic Encoder for Rotor Position Detection Based on Arbitrary Phase-Shift Phase-Lock Loop. *IEEE Trans. Ind. Electron.* **2022**, *69*, 2033–2043. [\[CrossRef\]](#)
25. Chen, S.; Ding, W.; Hu, R.; Wu, X.; Shi, S. Sensorless Control of PMSM Drives Using Reduced Order Quasi Resonant-Based ESO and Newton–Raphson Method-Based PLL. *IEEE Trans. Power Electron.* **2023**, *38*, 229–244. [\[CrossRef\]](#)
26. Allegro MicroSystems. Minimizing Angle Error in 2D Magnetic Sensors by Nonorthogonality Correction. Application Note AN126. 2023. Available online: https://www.allegromicro.com/-/media/files/application-notes/an126-minimizing-angle-error-by-orthogonality-correction.pdf?sc_lang=en (accessed on 10 January 2026).
27. Tran, T.N.C.; Nguyen, H.X.; Park, J.W.; Jeon, J.W. Improving the accuracy of an absolute magnetic encoder by using harmonic rejection and a dual-phase-locked loop. *IEEE Trans. Ind. Electron.* **2019**, *66*, 5476–5486. [\[CrossRef\]](#)
28. Kuntz, S.; Dauth, R.; Gerlach, G.; Ott, P.; Fella, S. Harmonic analysis of the arctangent function regarding the angular error introduced by superimposed Fourier series for application in sine/cosine angle encoders. *Sens. Actuators A* **2022**, *344*, 113585. [\[CrossRef\]](#)
29. Scikit-Learn Developers. MinMaxScaler—Scikit-Learn 1.8.0 Documentation. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html> (accessed on 10 January 2026).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.