Received December 7, 2020, accepted December 30, 2020, date of publication January 25, 2021, date of current version February 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054362

A Linear Compensation Method for Improving the Accuracy of an Absolute Multipolar Magnetic Encoder

JAE WAN PARK, HA XUAN NGUYEN, THUONG NGOC-CONG TRAN, AND JAE WOOK JEON, (Senior Member, IEEE)

Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 440-746, South Korea

Corresponding author: Jae Wook Jeon (jwjeon@yurim.skku.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIP), South Korea, through the G-ITRC Support Program supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2018- 20150-00742.

ABSTRACT This paper proposes a linear compensator algorithm to improve the performance of an absolute multipolar magnetic encoder (AMPME). An AMPME is an absolute magnetic rotary encoder that uses a multipolar magnet (MPM) to increase the resolution. The resolution can be dramatically increased in proportion to the number of poles in the MPM. However, various hardware problems that occur during the encoder manufacturing process degrade the AMPME performance. Also, harmonic components occur in the raw data due to various problems, such as the resistance error of the analog circuit, magnetic field overlap between the magnets, position error between the sensor and magnet, and pole-pitch difference. In particular, during the magnetization process of the MPM, the pole-pitch difference becomes a problem when the sizes of each pole are not uniform. This problem causes harmonic components that reduce the absolute position accuracy. To solve these problems, this paper proposes a linear compensation method. The proposed linear compensator consists of two parts. The first part is the enhanced ratiometric linearization for phase calculation and calibration. The second is the phase compensator for removing the phase difference via the pole-pitch difference of the MPM. The linear compensator improves various parameters by precomputing the offset, amplitude, and phase corrections. After compensating for sinusoidal signals, the linear compensator applies appropriate parameters at the appropriate times. This method is faster, easier to set up, and more accurate than the conventional method. Furthermore, this method is experimentally verified against the existing harmonic rejection method. Experimental results are provided to verify the effectiveness of the proposed method.

INDEX TERMS Absolute magnetic encoder, calibration, harmonic error, linearization, pole-pitch difference, multipolar magnet, phase locked loop.

I. INTRODUCTION

An absolute encoder provides an absolute rotation position. Unlike incremental encoders, an absolute encoder does not require a home sensor or limit sensor to identify the reference position. Additionally, the absolute encoder always remembers the physical position without using a battery. In the case of incremental encoders without a battery, the encoder is not able to locate its own position when the power is turned off and then on again. With an absolute encoder, it is possible to reduce overall battery management, making it particularly useful in systems where it is difficult to replace batteries. For example, robots hanging from the ceiling of an assembly line and underwater robots have batteries that are difficult to replace.

Encoders can be divided into optical and magnetic types according to the detection method. Although the optical encoder has high accuracy, its structure is complicated. Optical encoders use light-emitting elements placed in parallel and light-receiving elements placed to face light-emitting elements. An optical disk is placed between the light-emitting element and the light-receiving element. The signal from the

The associate editor coordinating the review of this manuscript and approving it for publication was Huiqing Wen.

parallel light-receiving element is binary encoded according to the position of the detection shaft. The performance of the optical encoder is determined by the precision of the optical disc. Its performance is significantly reduced if oil or dust penetrates the encoder system and covers code area of the optical disc. Magnetic encoders are structurally simple and cheaper than optical encoders because they only need to be placed so that the bipolar magnet (BPM) and magnetic-sensor face each other. Furthermore, the magnetic encoders do not affect the magnetic field even if dust or oil penetrates the system. However, the precision of the magnet is not sufficient, and the magnetic sensor is affected by noise. The performance of the magnetic encoder is smaller than the optical encoder. To compensate for this, both BPM and a multipolar magnet (MPM) can be added on the detection shaft to increases dramatically [1]–[4].

The absolute multipolar magnetic encoder's (AMPME) detection shafts are equipped with both an MPM and BPM. In this case, the AMPME can divide one revolution (of 360 degrees) by the number of poles in the MPM. In other words, since one revolution is expressed by multiple quadrature sinusoid signals, the resolution increases. However, the AMPME has hardware problems that impair its performance. These hardware problems cause harmonic components that differ from the main frequency in the quadrature sinusoid signals. First, the BPM and MPM are close to each other. Therefore, a magnetic field overlap occurs in each magnetic sensor. This makes the harmonic component similar to the period of the BPM input signal in the MPM signal measurements. Secondly, the harmonic component caused by the pole-pitch difference of the MPM is the biggest factor in deteriorating the absolute position accuracy. Additionally, tolerance issues may occur when assembling the encoder, causing misalignment between the sensor and magnet. Such misalignment creates differences in the amplitude and offset between the quadrature sinusoid signals. Because the harmonic components, amplitude difference, and offset difference will degrade the absolute position accuracy of the encoder, the AMPME requires compensation and calibration methods.

The following hardware techniques to reduce harmonic components have been studied. First, a magnetic shield can be added between the magnets to prevent the field from overlapping [3], [5]. Second, by increasing the number of sensors and placing them at intervals of equal length, can obtain the average error of each position [6], [7]. However, these methods did not completely remove harmonic components. Further, this method is unsuitable for a magnetic encoder that emphasizes a low-cost simple structure. Harmonic components can be reduced by supplementing the hardware, but this is not sufficient compared to the increase in complexity.

Therefore, many studies utilized signal processors to design digital compensators. The signal processor must calculate the phase of quadrature sinusoids that are output from the magnets and magneto resistor (MR) sensor to determine the rotation angle. This process requires noise reduction, calibration, and phase compensation. First, the signal processor can use the arctangent function to calculate the phase. To implement an arctangent function using a digital signal processor (DSP), an approximation is computed using a Taylor series polynomial. In contrast, the coordinate rotational digital computer (CORDIC) algorithm performs vector rotations in a two-dimensional plane and has a regular structure. To reduce the processing time, a modified vector rotational CORDIC (MVR-CORDIC) algorithm was studied [8]. The MVR-CORDIC reduces the number of iterations by 50% using various methods, including ''skip some microrotation angles'' and ''repeat some microrotation angles'' while maintaining good performance. However, this arctangent method is not suitable for high-precision encoders because it only calculates the phase without countermeasures against phase errors from the hardware. Also, note that parallelization using an FPGA can speed up the process dramatically, but the cost of the encoders will increase significantly. The phase-locked loop (PLL) method is very effective for calculating phases, as well as compensating the phase and removing random noise. Additionally, the computational complexity is low compared to the arctangent method, making it suitable for detecting phases in the encoder. Therefore, various studies have been performed to detect the position using a PLL [10]–[16]. First, there are studies that accurately and quickly calculate phases via the quadrature sinusoid (sine and cosine) outputs from the MR sensors of the AMPME [10]–[12]. The studies [10] and [11] present a PLL method for detecting the phases of quadrature sinusoid signals in place of the arctangent method for DSP. This approach includes methods for accuracy improvement and calibration. The PLL method is more efficient for calculating the phase and removing random noise than the arctangent method. However, it cannot eliminate harmonic components close to the main frequency band. Further, a PLL is more effective for incremental encoders or resolvers than absolute position detection because it is most effective for periodic signals with many samples [13]–[16]. Alternatively, there was a study to find the absolute position by measuring the magnetic field of a permanent-magnet synchronous motor (PMSM) without an encoder [12]. For the PLL case, the encoder can achieve more accurate results when there are more quadrature sinusoid input samples. The PLL can improve the resolution by applying the input signal and time pulse, thus recovering the phase error as time passes. The study [16] analyzed a technique that effectively removes random noise, DC offset, phase shift, and the waveform distortion of quadrature sinusoids output from a resolver in a linear motor, thus improving the resolution using the advanced adaptive digital phase-locked loop (AADPLL) algorithm. Furthermore, when applied to a DSP, this algorithm was very effective. However, this study does not consider hardware-based harmonic components. Additionally, the AMPME is an absolute position encoder, so it is necessary to detect the exact position even if no sample has been input in advance. In other words, the physical error must be compensated for without any previously

input data. This is directly related to the problem of eliminating harmonic components in the AMPME. Moreover, the PLL cannot reduce harmonic components due to the pole-pitch differences and magnetic field overlap. In addition, if the PLL parameters are set incorrectly, there may be a delay time or repeatability error when the speed of the detection shaft changes. Therefore, to use the signal processing algorithm of AMPME in industrial sites, the following three requirements must be satisfied. First, the AMPME should be to improve the accuracy of the absolute position. For this, it is necessary to remove the harmonic components with frequencies other than the main frequency. Second, the processing algorithm should be optimized to reduce the computation time. The advantage of the AMPME used in industrial sites is that it has good performance at a low cost. In other words, the algorithm should use low-cost processors. Third, the initial setting of the signal processing algorithm should be easy. When making an AMPME, if parameters to sets are many and complicated, this can increase the cost and decrease the stability. And, depending on the parameters to be set, performance may be significantly affected.

If we compare the existing studies, the arctangent method cannot compensate for the harmonic components. Therefore, it is difficult to apply to systems that require harmonic compensation, such as AMPME. The PLL-based methods increase in complexity according to the performance requirements and an increase in the number of parameters to be set. Therefore, if there are high performance requirements, it is necessary to use a high-performance processor. To satisfy all of the requirements mentioned above, we present a simple, easy to set up, and accurate linear compensator.

This paper focuses on the patterns of all harmonic components and their periodicity. Hence, the encoder stores the phase size, amplitude, and offset at every cycle of the MPM signal during one revolution, and applies these parameters at the appropriate times. To implement this, we propose a linear compensator. This compensator consists of a phase compensator and a phase detector that enhances the existing ratiometric linearization (RL) technique [18].

1) Enhanced ratiometric linearization: This paper presents a method to remove the random noise, offset, and amplitude differences by adding a low pass filter (LPF) and a static calibrator to the ratiometric linearization process.
2) Phase compensator: The AMPME stores each phase size for all cycles in a look-up table (LUT) and resizes the position area of each MPM pole to compensate for any position errors.

This linear compensator effectively reduces harmonic components while using few computational resources and an LUT of word size $N_p \times 18$ (N_p is the number of poles in the MPM).

This paper is organized as follows: Section II provides an overview, an analysis of non-idealities, and the problems associated with the PLL. Section III explains the enhanced RL. Section IV explains the phase compensator. Section V presents the experimental results. Finally, Section VI concludes this study.

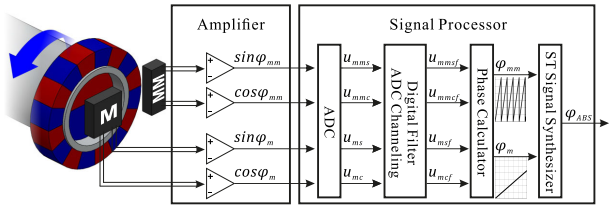

FIGURE 1. Block diagram of the improved resolution via MPM.

II. OVERVIEW AND RELATED WORK

A. OVERVIEW OF THE AMPME: IMPROVED RESOLUTION

The AMPME can increase the resolution according to the number of poles in the MPM [1]–[4]. Fig. 1 shows a block diagram of the encoder that increases the resolution by adding an MPM to the shaft. When the tunneling MR (TMR) sensor is placed on the opposite side of the BPM and the anisotropic MR (AMR) sensor is placed on the same side of the MPM, the TMR sensor outputs one cycle for the quadrature sinusoid signals. Additionally, the AMR sensor outputs the same number of quadrature sinusoid signals as the number of poles of the MPM over a single rotation of the shaft. The signal φ_{mm} is the phase of the MPM and directly corresponds to the resolution. The signal φ_m is the phase of the BPM; it serves as a reference to find the cycle number of the signal φ_{mm}. Thus, the AMPME uses a 12-bit analog to digital converter (ADC) and a 24-pole MPM, AMPME can obtain a resolution of 98304 (360/98304 = 0.003662[deg]). The high-resolution absolute position of the AMPME can be written as follows:

$$\varphi_{abs} = \frac{\varphi_{mm} + 2\pi \times n}{N_p} \tag{1}$$

where φ_{abs} is the absolute position of the AMPME, φ_{mm} is the phase of each quadrature sinusoids obtained from the MPM, n is the cycle number of φ_{mm}, and N_p is the number of poles of the MPM. The range of n is from 0 to $(N_p - 1)$ and is calculated as:

$$n = \left\lfloor \frac{\varphi_m}{2\pi / N_p} \right\rfloor \tag{2}$$

where $2\pi/N_p$ is a unit that indicates the phase size of each pole of the MPM. After determining the current cycle number n, the AMPME can find the absolute position of the shaft by multiplying n by the phase size $2\pi/N_p$ and adding the current φ_{mm}. In practice, the output phase of the MPM contains various disturbances, which are analyzed in the next section. Therefore, the absolute position (1) should take into account these errors.

B. ANALYSIS OF NON-IDEALITIES

Physical factors that hinder the performance of the AMPME include tolerance, magnetic field overlap, and pole-pitch difference.

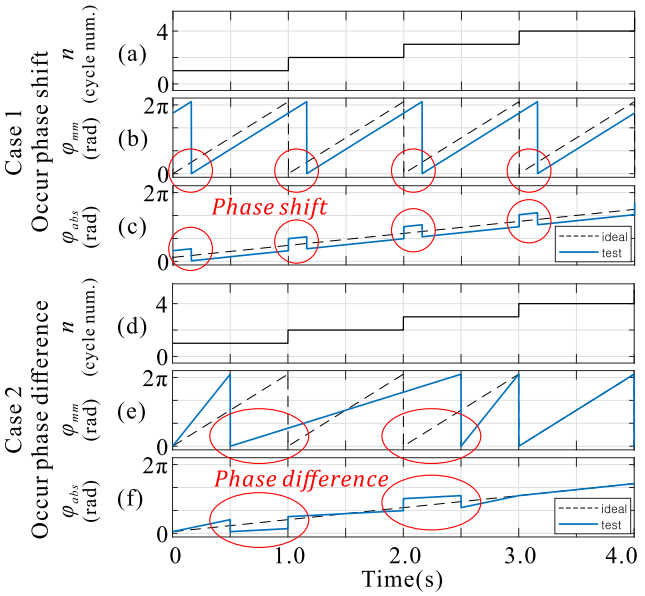

FIGURE 2. Cycle number mismatch. The red circles shown the cycle number mismatch. (a) and (d) are graphs of the cycle number n calculated by the index calculator with φ_m. (b) is a graph of φ_{mm} with a phase shift. (e) is a graph of φ_{mm} with a phase difference. (c) and (f) are φ_{abs} with cycle number mismatch relative to (b) and (e).

1) TOLERANCE

The tolerance creates amplitude differences, offsets, and distortions in the quadrature sinusoid signals. Tolerance can be divided into electrical tolerance and assembly tolerance. First, electrical tolerance occurs between the MR sensor and the ADC in the encoder signal processor. The encoder amplifies the signal using an operational amplifier (OP-Amp) between the MR sensor and the ADC. At this time, amplitude and offset errors occur due to the resistance error of the OP-Amp gain and the line resistance of the circuit. Additionally, every electronic system has random noise. Magnet and sensor positional errors result in decreased assembly tolerance during encoder assembly processes. Secondly, the distance error between the magnet and the sensor results in the amplitude and offset errors. The magnets and sensors are arranged to face each other, and sinusoidal distortions occur if the center points are not aligned with each other. Further, when assembling the BPM and MPM on the shaft, cycle number mismatch occurs if the initial positions of the poles of the two magnets are not synchronized. This is the phase shift of φ_{mm}. Fig. 2(c) shows what errors occur in φ_{abs} according to phase shift.

2) THE POLE-PITCH DIFFERENCE

The pole-pitch difference of an MPM is caused by differences in the intervals of poles. These imbalances cause the sizes of each period of φ_{mm} within one revolution to be different. First, the sizes of n and φ_{mm} are different, resulting in a cycle number mismatch. Fig. 2(e) is a graph with a phase difference added. Therefore, a new cycle number calculation method is needed to solve this mismatch problem. In section IV, we present a new index (cycle number) calculator to be

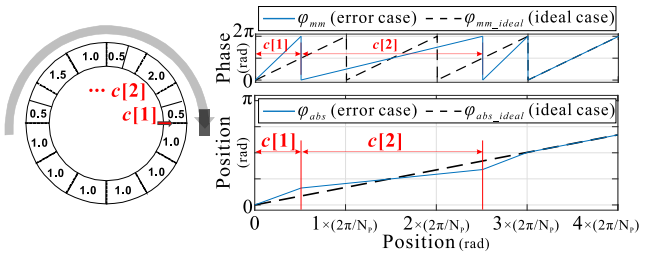

FIGURE 3. The pole-pitch difference of MPM, The $c[n]$ is the ratio of the phase according to the pole-pitch difference. (Suppose that shaft rotate in constant speed.)

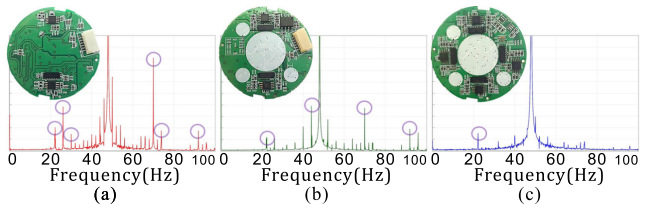

FIGURE 4. Encoder signal processing systems with a varying number of MR sensors (1, 2, and 4 MR sensors).

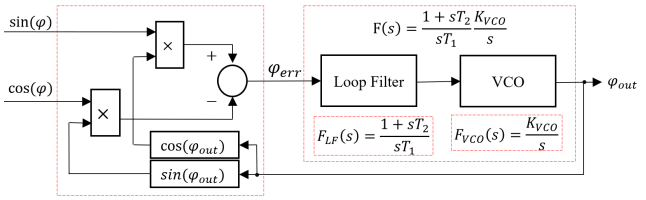

FIGURE 5. Conventional PLL Idea.

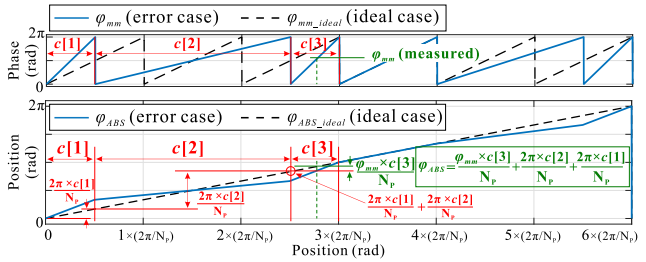

FIGURE 6. Calculation of the ideal absolute position. φ_{mm} is signal of AMPME's MPM. φ_{abs} represents the ideal absolute position when the current position is at the third pole of MPM.)

implemented in the phase compensator. Even if we solve the abovementioned cycle number mismatch problem, the phase difference is still a big problem. Furthermore, this is the main harmonic component that degrades the performance of the AMPME. Fig. 3 shows an example where this problem is significant. The graph shows that different sizes were measured among signal cycle. For each ideal pole-pitch ratio 1, $c[1]$ is change by 0.5 times and $c[2]$ by 2 times. This means that the pole-pitch size is larger or smaller than the reference. The situation in, Fig. 3 assumes that the detection shaft moves at a constant velocity. When the signal output from the sensor is collected by the Signal Processor, the interval of each phase may be different, as shown in the phase graph. Furthermore, if the signal processor uses this signal to calculate the absolute position, it will get a nonlinear result, like the position graph.

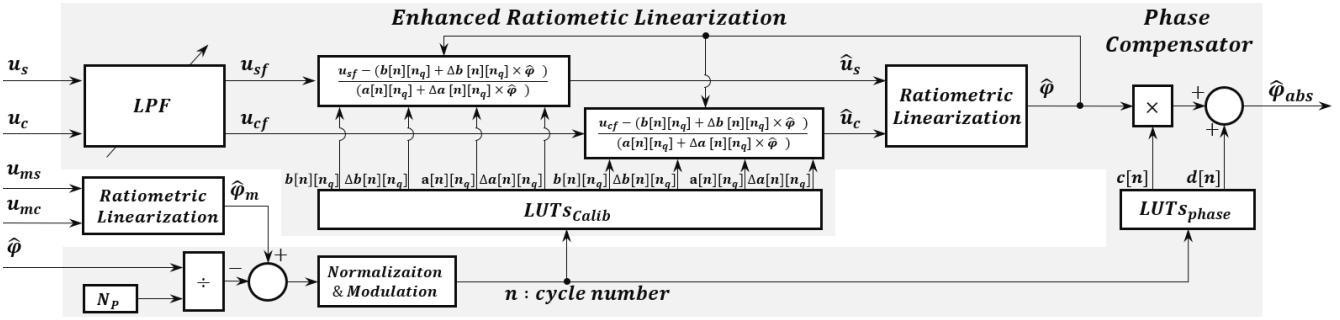

FIGURE 7. The proposed linear-compensator. u_s, u_c are sine, cosine signal of MPM. u_{ms}, u_{mc} are sine, cosine signal of BPM.

The obtain aim of this paper is to solve this problem. By applying the hardware method, an increasing number of MR sensors is used to mitigate the effects of the harmonic components, as shown in Fig. 4. However, this method is necessarily costly. Therefore, this paper includes digital and hardware methods to improve the accuracy of the AMPME using two magnetic sensors, as seen in Fig. 4(b). Also, the output signals can be rewritten as follows:

$$u_s = a_s \sin(\varphi) + b_s + e_s \tag{3}$$
$$u_c = a_c \cos(\varphi) + b_c + e_c \tag{4}$$

where a_s and a_c are the amplitudes of the quadrature sinusoids inputs, b_s and b_c are the offsets, e_s and e_c are random noise, and φ_{mm} is the output phase of the MPM. However, the problem related to the pole-pitch difference is not seen in u_s and u_c. The reason is that the ranges of angles for φ_{mm} are each different. Thus, the error does not appear in φ_{mm} after only one period. Therefore, we redefine φ_{abs} again as follows, taking into the account phase difference.

$$\varphi_{abs} = \frac{\varphi_{mm} \times c[n]}{N_p} + \sum_{i=1}^{n-1} \left\{ \frac{2\pi \times c[i]}{N_p} \right\} \tag{5}$$

where $c[n]$ is the ratio of phase to pole-pitch difference. This is the ratio of actual pole-pitch size and ideal signal depending on pole-pitch difference. The previous phase accumulates according to the n-th cycle timing. The φ_{abs} formula in Fig. 6 shows the pole of MPM in the 3rd cycle timing. Therefore, the biggest difference in the proposed method compared to previous research is that it considers equation (5) to eliminate errors due to pole-pitch difference.

C. RELATED WORK

Fig. 5 shows this conventional PLL (CPLL) method for an absolute magnetic encoder (AME). The advantage of a PLL is that it can detect the phase in place of the arctangent functions. And, PLL can compensate signal.

1) HARMONIC ERROR

A technique for eliminating harmonic errors in the MPM has been previously researched [12]. This method utilizes technology that can detect the magnetic field signal of a permanent-magnet (PM) synchronous motor (PMSM) as well as its position. This approach can eliminate the harmonic error generated in the magnetic field signal measured by each hall sensor, thereby achieving more accurate position control. However, that research does not mention the phase differences that can occur when the spacing between the magnets is not constant.

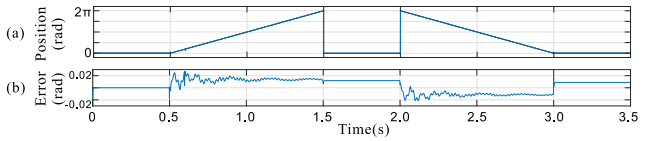

FIGURE 8. Simulation of CPLL with ideal inputs (without harmonic errors). (a) is the ramp-input (value:position) and (b) is the error of the CPLL result.

2) DC ERROR

DC error means that the position results of the encoder are constantly different from the actual value. This is mainly caused by input (raw data) measurement delay, signal processing for compensation, and communication delay. Fig. 8 shows errors when an ideal frequency step signal is input to the AMPME using a PLL. As shown in Fig. 8, the CPLL cannot remove the DC error, and has further issues mitigating the absolute position errors in the stopped state. The phase error of the PLL system can be written as:

$$\varphi_{err}(s) = \frac{s^2}{s^2 + \frac{K_{VCO}T_2}{T_1}s + \frac{K_{VCO}}{T_1}} \varphi(s) \tag{6}$$

Using the final-value theorem, the steady-state errors of the CPLL in the phase step ($\varphi(s) = \triangle\varphi/s$), frequency step ($\omega/s^2$), and frequency ramp ($\gamma/s^3$) are as follows:

$$\lim_{s\to 0} \varphi_{err}(s) = \frac{s^2}{s^2 + \frac{K_{VCO}T_2}{T_1}s + \frac{K_{VCO}}{T_1}} \frac{\triangle\varphi}{s} = 0 \tag{7}$$
$$\lim_{s\to 0} \varphi_{err}(s) = \frac{s^2}{s^2 + \frac{K_{VCO}T_2}{T_1}s + \frac{K_{VCO}}{T_1}} \frac{\omega}{s^2} = \frac{1}{\frac{K_{VCO}}{T_1}} \tag{8}$$
$$\lim_{s\to 0} \varphi_{err}(s) = \frac{s^2}{s^2 + \frac{K_{VCO}T_2}{T_1}s + \frac{K_{VCO}}{T_1}} \frac{\gamma}{s^3} = \infty \tag{9}$$

From (8), (9), it is clear that the phase output of the CPLL contains DC errors related to the frequency step and frequency ramp. Therefore, based on the operating characteristics of AMPMEs, the DC error results in encoder position errors.

This paper presents a linear compensation method that includes an enhanced ratiometric linearization and compensation algorithms to overcome the PLL problem and improve the accuracy of the AMPME, as shown in Fig. 7.

III. ENHANCED RATIOMETRIC LINEARIZATION

This section describes enhanced ratiometric linearization (Enhanced-RL). Since we propose a method to replace the PLL, a new method to calculate the phase of the quadrature sinusoid signals is necessary. Therefore, we improved the performance by adding a filter and calibrator to the RL. Therefore, the Enhanced-RL method performs random noise filtering, calibration, and phase detection.

A. PRE-FILTER

The Enhanced-RL eliminates random noise using only a very simple first-order LPF in order to reduce the computation time and minimize the DC error. Further, the use of a built-in filter in the ADC module of the signal processor is sufficient to counter random noise. Therefore, u_{sf} and u_{cf} in Fig. 7 are the results of filtering u_s and u_c with the LPF. After that, the Enhanced-RL gets a signal from which the random noise has been removed.

B. STATIC CALIBRATOR

The static-calibrator calibrates amplitudes and offsets of quadrature sinusoid signals of all cycles within a revolution (360 degrees). The basic idea of the static-calibrator is to calibrate it to each of the parameters for each position. This is done because the maximum and minimum values of quadrature sinusoid signals for each position in one revolution are different for each pole of the MPM. In other words, due to electrical and mechanical distortions, the inputs u_s and u_c have variable amplitudes and offsets for each cycle. Therefore, (3) and (4) can be modified as follows:

$$(3) \longrightarrow \sin(\varphi) = \hat{u}_s = \frac{u_s - \hat{b}_s}{\hat{a}_s} \tag{10}$$

$$(4) \longrightarrow \cos(\varphi) = \hat{u}_c = \frac{u_c - \hat{b}_c}{\hat{a}_c} \tag{11}$$

where $\hat{a}_s$ and $\hat{a}_c$ are the amplitudes of each position, $\hat{b}_s$ and $\hat{b}_c$ are the offsets of each position, and e_s and e_c are random noise. Additionally, the Enhanced-RL approach uses u_{sf} and u_{cf} that has the random noise removed. When the encoder already knows the offset and amplitude in advance for each cycle of u_{sf} and u_{cf}, the encoder can calibrate the quadrature sinusoid signals. Therefore, the purpose is to precompute and store different amplitudes and offsets for each position. Thus, to calculate each parameter, the AMPME measured the signals u_{sf} and u_{cf} while rotating the encoder shaft at a constant frequency. The amplitudes and offsets of the input signals are calculated based on the maximum and minimum points, as shown Fig. 9, 11. This method is presented as follows:

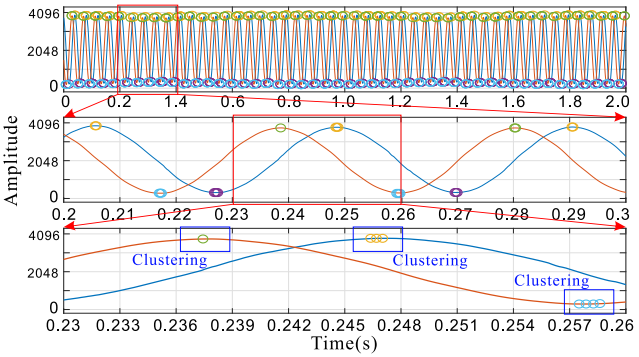

FIGURE 9. Maximum and minimum points.

1) DETECT THE ZERO GRADIENT POINT

AMPME detects the maximum and minimum points of each cycle while the shaft rotates at a constant frequency. The moment when the slope of the quadrature sinusoid signals becomes 0 (zero-gradient) at a constant frequency is a maximum or minimum point. The following formula is the zero-gradient for detecting that point.

$$zero\ gradient, \left|\nabla u_{sf,cf}\right| = 0 \tag{12}$$

The sum of the left slope and the right slope is 0 based on the current position; this position can be defined as a zero-gradient. Thus, (12) is defined again, as follows.

$$(12) \longrightarrow \left|(u_{sf,cf}(n_s+1) - u_{sf,cf}(n_s-1))\right| = 0 \tag{13}$$

where n_s is the sampling number of the processing.

$$\begin{cases} store, & |(u_{sf,cf}(n_s+1) - u_{sf,cf}(n_s-1))| < \beta_2 \\ skip, & |(u_{sf,cf}(n_s+1) - u_{sf,cf}(n_s-1))| \geq \beta_2 \end{cases} \tag{14}$$

Therefore, if the gradient value is lower than the threshold β_2, as described above, the position is temporarily stored as the maximum or minimum point. The threshold β_2 is defined as follows:

$$\beta_2 = \beta_1 A \left|1 - \cos(N_p \frac{2\pi}{wT_s})\right| \tag{15}$$

where β_2 is the threshold and the slope between the maximum (or minimum) point and the next point in the quadrature sinusoid, N_p is the number of poles in the MPM, w is the frequency, and T_s is the sampling time. Thus, $\cos(N_p \frac{2\pi}{wT_s})$ represents the next sample point immediately after the maximum point of the quadrature sinusoid. The next sample point can be defined using w and T_s. In the case of an AMPME, N_p poles are required since the number of poles increases the quadrature sinusoid cycles. Thus, $\left|1 - \cos(N_p \frac{2\pi}{wT_s})\right|$ is the difference between the maximum point and the next sample

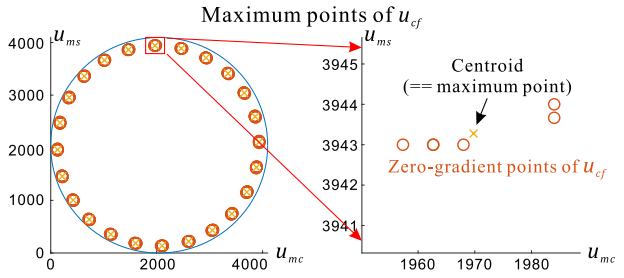

FIGURE 10. The maximum point of u_{cf} by K-means algorithm.

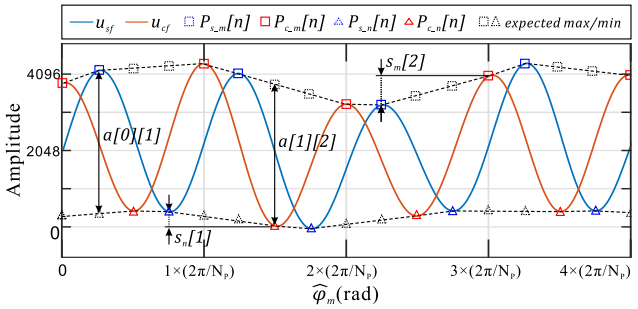

FIGURE 11. Measured amplitude line and offset line. (a) is the amplitude line of sine and. (b) is the amplitude line of cosine.

point in the quadrature sinusoid. Multiplying this term by the amplitude A converts it to the range of input signals. Finally, β_1 is weight. As β_1 increases, the detection range of the maximum and minimum points becomes wider. However, if β_1 is too small, no points can be detected. In this paper, β_1 was set to 3 for stability. Fig. 9 shows the detected maximum and minimum points. And, this method detects multiple maximum or minimum points in one location. So it needs clustering them and calculate the centroid of each zero-gradient group.

2) FIND REFERENCE POINTS OF EACH CLUSTER USING K-MEANS

The purpose of this step is to calculate the maximum and minimum points of each cycle by clustering the measured zero-gradient points of maximum and minimum positions and finding their centroid, as shown in Fig. 9. At this time, the centroid must be detected based on the reference signal $\hat{\varphi}_m$ because the absolute position uses $\hat{\varphi}_m$ to separate each cycle of u_s and u_c. In this paper, we use the simple k-means algorithm. If we draw a Lissajous based on $\hat{\varphi}_m$ and list the maximum (or minimum) of each cycle, we can find the centroid of each maximum (or minimum) points using the k-means algorithm. Fig. 10 shown the Lissajous of the quadrature sinusoid and the results of the k-means algorithm.

3) CALCULATE THE AMPLITUDES AND OFFSETS

Last, using the detected maximum and minimum points, the amplitude and offset are calculated for each cycle of $\hat{\varphi}$ and then stored in the LUT. At this time, because u_s and u_c are quadrature sinusoid signals and output from one sensor at the same time, the change of amplitude and offset by the magnetic field are same. Therefore, the maximum and minimum points of the quadrature sinusoid signals are used together to calculate the common amplitude and offset. By applying the characteristics of quadrature sinusoid signals, one cycle can be divided into four parts. Fig. 11 shows the maximum and minimum points divided into four parts for each cycle. There are two maximum and minimum points in a cycle. Therefore, the other two are predicted by creating slopes based on nearby points. Using this method, we can store the amplitude and offset of each period in advance by calculating the following:

$$a[n][n_q] = \begin{cases} P_{c\_m}[n] - (P_{c\_n}[n] - \frac{2}{3}s_m[n]), & n_q = 0 \\ P_{s\_m}[n] - (P_{c\_n}[n] - \frac{1}{3}s_m[n]), & n_q = 1 \\ (P_{s\_m}[n] - \frac{1}{3}s_n[n]) - P_{c\_n}[n], & n_q = 2 \\ (P_{s\_m}[n] - \frac{2}{3}s_n[n]) - P_{s\_n}[n], & n_q = 3 \end{cases} \tag{16}$$

$$b[n][n_q] = \begin{cases} \frac{1}{2}\left(P_{c\_m}[n] + (P_{c\_n}[n] - \frac{2}{3}s_m[n])\right), & n_q = 0 \\ \frac{1}{2}\left(P_{s\_m}[n] + (P_{c\_n}[n] - \frac{1}{3}s_m[n])\right), & n_q = 1 \\ \frac{1}{2}\left((P_{s\_m}[n] - \frac{1}{3}s_n[n]) + P_{c\_n}[n]\right), & n_q = 2 \\ \frac{1}{2}\left((P_{s\_m}[n] - \frac{2}{3}s_n[n]) + P_{s\_n}[n]\right), & n_q = 3 \end{cases} \tag{17}$$

where $a[n][0], \ldots, a[n][3]$ are the amplitude of $n-th$ cycle. $b[n][0], \ldots, b[n][3]$ are the offset of the n-th cycle. $P_{s\_m}[n]$ and $P_{c\_m}[n]$ are the maximum points of the $n-th$ cycle's quadrature sinusoid. $P_{s\_n}[n]$, and $P_{c\_n}[n]$ are the minimum points of the $n-th$ cycle's quadrature sinusoid. s_m and s_n are temporary values for calculating the predicted value without maximum and minimum points, as follows.

$$\begin{cases} s_m[n] = P_{c\_m}[n+1] - P_{s\_m}[n] \\ s_n[n] = P_{c\_n}[n] - P_{s\_n}[n-1] \end{cases} \tag{18}$$

where since each cycle is repeated according to the rotation of the MPM, the cycle before returned 0-th is the $(N_p - 1)$-th cycle.

These factors are used to pre-calculate and store the amplitude and offset values of 4 parts for each cycle. So, static-calibrator is a method of calibrating amplitudes and offsets with parameters stored in advance for each cycle. The method is as follows.

$$\hat{a} = \left(a[n][n_q] + \bigtriangleup a[n][n_q] \times (r_q - n_q)\right) \tag{19}$$

$$\hat{b} = \left(b[n][n_q] + \bigtriangleup b[n][n_q] \times (r_q - n_q)\right) \tag{20}$$

$$r_q = 4 \times \frac{\hat{\varphi}[n_s - 1]}{2\pi} = \frac{\hat{\varphi}[n_s - 1]}{\pi/4} \tag{21}$$

$$n_q = \lfloor r_q \rfloor \tag{22}$$

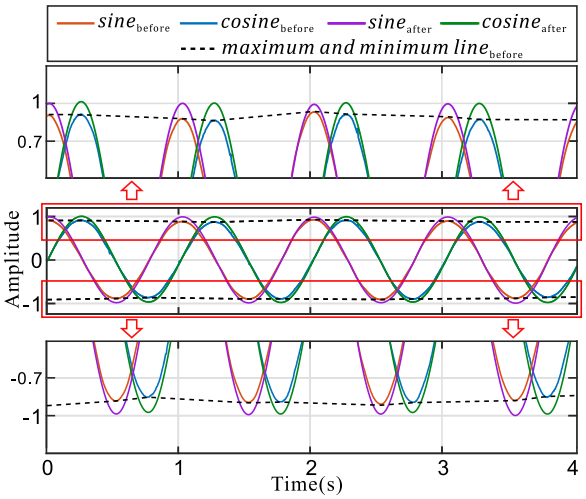

FIGURE 12. Simulation results of the static calibrator.

where $\hat{a}$ is the amplitude that calibrated by each position. $\hat{b}$ is the offset that calibrated by each position. As mentioned above, the sine and cosine signals are output from the same sensor. And, the sensor is affected by the same magnetic field, so the amplitude and offset are the same ($\hat{a}_s = \hat{a}_c = \hat{a}$ and $\hat{b}_s = \hat{b}_c = \hat{b}$). r_q is the ratio between each cycle of quadrature sinusoids. n_q is the index number among then the four parts of the quadrature sinusoid. We can simply get n_q by using r_q. $\hat{\varphi}[n_s - 1]$ is the MPM phase after applying Enhanced-RL (the difference is very small even if we use the results from the previous sample).

$\triangle a[n][n_q]$ is the deviation between $a[n][n_q]$ and $a[n][n_q - 1]$. r_q is the ratio of $\hat{\varphi}[n_s - 1]$, which ranges from $[0, 4]$, the closer it is to the end of the current cycle. Therefore, $r_q - n_q$ is inter distance between of $n[n][nq]$ and $n[n][n_q - 1]$ as expressed by ratio [0, 1]. This means the AMPME can calculate maximum and minimum lines. Thus, the AMPME can calculate and apply the exact amplitude and offset for every position. In addition, storing $\triangle a[n][n_q]$ in the LUTs can reduce the computational complexity and correct the signal. Additionally, $\triangle b[n][n_q]$ is the offset deviation and has the same calculation method as the amplitude deviation. Fig. 12 shows the result of a static calibrator.

C. RATIOMETRIC LINEARIZATION

After filtering and calibrating using the LPF and static calibrator, the input sine and cosine signals are normalized to the range $[-1, 1]$. Therefore, the triangular wave directly computed from $(\hat{u}_s, \hat{u}_c)$ can be written as:

$$V_1(\hat{\varphi}) = \frac{|\hat{u}_s| - |\hat{u}_c|}{|\hat{u}_s| + |\hat{u}_c|} \tag{23}$$

while the ideal triangular wave is calculated by:

$$V_{tri}(\hat{\varphi}) = \frac{4}{\pi}|\sin^{-1}(\hat{\varphi})| \tag{24}$$

The triangular wave V_1 created by formula (23) differs from the ideal triangular wave V_{tri} due to structural problems of

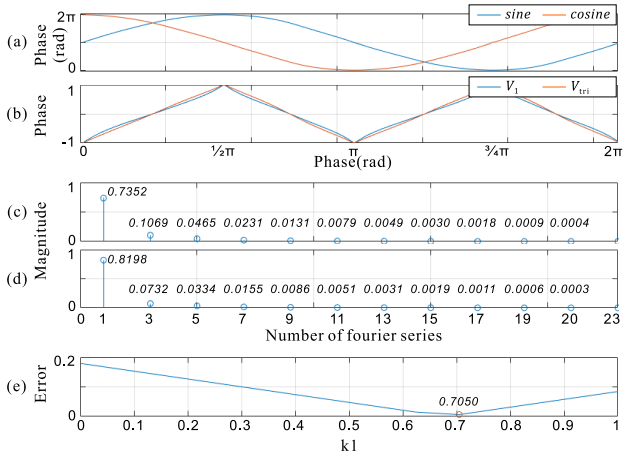

FIGURE 13. The parameters of the ratiometric linearization. (a) is the sinewave. (b) is V_1 and V_{tri}. (c) is the Fourier series a_n of V_1. (d) is the Fourier series b_n of V_{tri}. (e) is used to determine the $k1$ that gives the best efficiency when applied to $V2$.

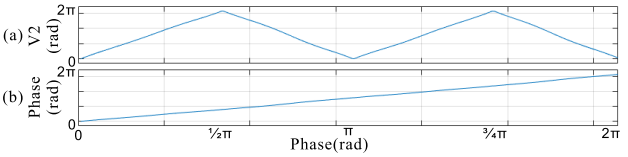

FIGURE 14. The simulation results of ratiometric linearization. (a) is the advanced triangular wave V_2. (b) is the results of ratiometirc linearzation using formula (28).

the sine wave (24). By calculating the difference between V_1 and V_{tri} and compensating for it, we can construct the compensated triangular wave V_2, as below:

$$V_2(\hat{\varphi}) = 2\pi(k_1 V_1(\hat{\varphi}) + k_2 \cos(2\hat{\varphi} - \pi) + 1) \tag{25}$$

$$V_2(\hat{\varphi}) = 2\pi\left(k_1 \frac{|\hat{u}_s| - |\hat{u}_c|}{|\hat{u}_s| + |\hat{u}_c|} + k_2 \frac{\hat{u}_s^2 - \hat{u}_c^2}{\hat{u}_s^2 + \hat{u}_c^2} + 1\right) \tag{26}$$

The parameters k_1 and k_2 can be determined based on the shape of the sine wave. Thus, we should design it to match the sine wave inputs of the hardware, then compute the Fourier series of the two triangular waves to compensate for the signal compared to the ideal triangular signal (Fig. 13(c) and (d)).

The last step is to find the value of k_1 to be applied to V_2 that is closest to the ideal triangle. The definitions of k_1 and k_2 are shown below, as described in a previous paper [18].

$$k_1 a_1 + k_2 = b_1 \tag{27}$$

First, k_1 is determined, and we then apply all possible k_1 values to determine the most efficient value. As shown in Fig. 13(e), we used a k_1 value of 0.7050 and a k_2 value of 0.3015 for the AMPME. (a_1 and b_1 are the first terms of the Fourier series, as shown Fig. 13.) From that, the output

phase can be obtained as:

$$\hat{\varphi} = \begin{cases} \dfrac{V_2(\hat{\varphi})}{4}, & (\hat{u}_s > 0)\ and\ (\hat{u}_c > 0) \\ \pi - \dfrac{V_2(\hat{\varphi})}{4}, & (\hat{u}_s > 0)\ and\ (\hat{u}_c < 0) \\ \pi + \dfrac{V_2(\hat{\varphi})}{4}, & (\hat{u}_s < 0)\ and\ (\hat{u}_c < 0) \\ 2\pi - \dfrac{V_2(\hat{\varphi})}{4}, & (\hat{u}_s < 0)\ and\ (\hat{u}_c > 0) \end{cases} \tag{28}$$

The output phase is directly calculated from the input signals ($\hat{u}_s$ and $\hat{u}_c$) via the triangular wave. Therefore, the characteristic DC error of the CPLL during the frequency step and frequency ramp is removed.

IV. PHASE COMPENSATOR

The phase $\hat{\varphi}$ of the MPM detected by Enhanced-RL has not yet removed the phase difference by pole-pitch difference. Also, the problem remains that the index n (i.e., the cycle number) must be applied to the LUTs of the Enhanced-RL and phase compensator with the correct timing. Therefore, this section describes the phase compensator and index calculator.

A. PHASE COMPENSATION METHOD

The phase compensator can mitigate the error caused by the phase difference in each signal cycle that results from the pole-pitch difference of the MPM. The phase of the cycle signal $\hat{\varphi}$ differs in each cycle when compared with the absolute position. The encoder pre-calculates and saves the ratio of the phase $c[n]$, as already mentioned in Sections II and Fig. 3, then compensates for the phase using (5). When calculating the absolute position of the AMPME, we can apply the pre-calculated ratio of the phase with the correct timing. Therefore, as shown in Fig. 15, we need to find the zero points $d[n]$ first. $c[n]$ can be calculated by $d[n]-d[n-1]$. Accordingly, (5) can be redefined as follows:

$$\hat{\varphi}_{abs} = c[n] \times \hat{\varphi} + d[n] \tag{29}$$

$$c[n] = \frac{d[n] - d[n-1]}{2\pi / N_p} \tag{30}$$

where $d[n]$ is the position in $\hat{\varphi}_m$ at the zero point when $\hat{\varphi}$ drops from 2π to 0 (i.e., the zero point of each $\hat{\varphi}$ cycle determines the position within the reference signal $\hat{\varphi}_m$). $c[n]$ is the ratio of each cycle of $\hat{\varphi}$ to the ideal signal cycle size. These values represent the phase differences that the encoder detects based on the physical location. Fig. 15 shows how $c[n]$ and $d[n]$ are defined in a signal, demonstrating an extreme example of the pole-pitch difference in the MPM. $2\pi/N_p$ is the interval of each cycle when the signal is ideal. Thus, we can find the slope $c[n]$ of the error signal using this information.

B. INDEX CALCULATOR

The LUTs, which store the parameters, are used to increase the performance of the linear compensator. Even if the LUTs are well made, their performance can vary depending on what

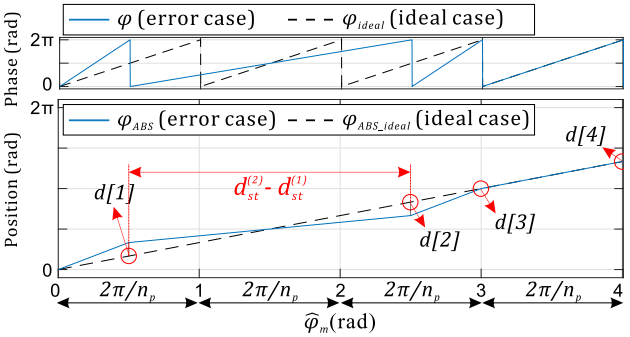

FIGURE 15. The parameters (**c[*n*]** and **d[*n*]**) in error signal.

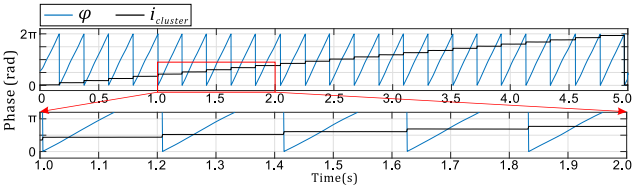

FIGURE 16. Simulation of index calculator. (a) is phase of MPM and $i_{cluster}$. (b) is an enlargement of (a).

data is selected at which moment from the cyclical signals of the MPM. Therefore, we designed an index calculator that applies these parameters in an accurate and timely match. The conventional method of detecting index is to distribute the reference signal $\hat{\varphi}_m$ equally to each part and compare the current value to each part, as in formula (2). However, this method has several problems. First, when the encoder is assembled, the signals of MPM and BPM cannot perfectly match the zero-position. Then, the interval of $\hat{\varphi}$ for each cycle is different due to the pole-pitch difference of the MPM. Fig. 2 and Fig. 15 demonstrate these problems. The proposed method in this paper is to compare $\hat{\varphi}_m$ and $\hat{\varphi}$ to return the current value to zero-points for each cycle at all times. This method has been studied in a multi-turn absolute encoder using gear systems [19][20].

$$i_{cluster} = nomal(\hat{\varphi}_m - \frac{1}{N_p}\hat{\varphi}) \tag{31}$$

$$n = \left\lfloor i_{cluster} / \frac{2\pi}{N_p} \right\rfloor \tag{32}$$

Here, N_p is the number of poles in the MPM, and the ratio of $\hat{\varphi}_m$ and $\hat{\varphi}$ differs depending on N_p. When AMPME subtracts $\hat{\varphi}/N_p$ from the current phase $\hat{\varphi}_m$, it is always possible to correct the phase to the zero-point. If the signal moves onto the next cycle, then $i_{cluster}$ will step up that like a step signal. However, this signal can out the range of 0 to 2π. Thus, normalization must use the limit function. Fig. 16 shows the $i_{cluster}$ signal obtained from (31). We can see that this process correctly matches the zero-position of $\hat{\varphi}$. The order of each cycle can be obtained as an index value using equation (32).

V. EXPERIMENTS

To implement the proposed ideas, we developed a 24-pole AMPME hardware and encoder signal processing system.

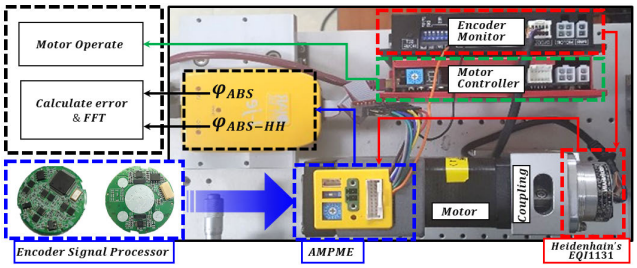

FIGURE 17. Experiment system.

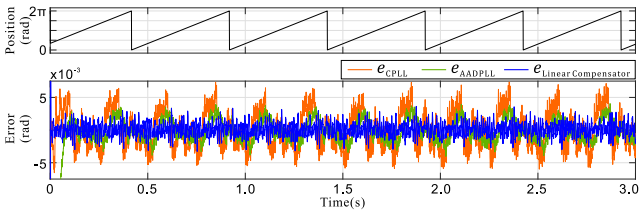

FIGURE 18. Results of the constant frequency case.

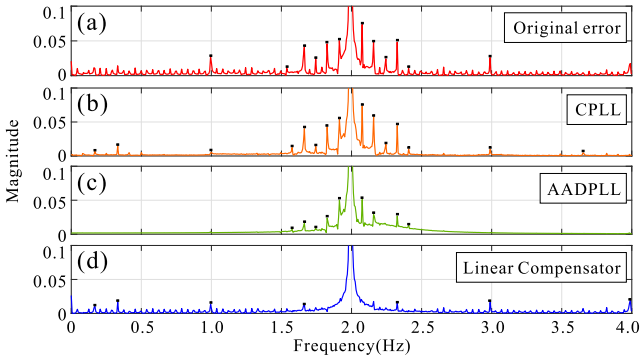

FIGURE 19. FFT analysis of results.

The processor is STM32F405RG (Cortex-M4F, 168MHz) and includes a 12bit ADC. Therefore, the encoder can process a resolution of up to 98304. To compare the accuracy of the proposed idea with that of a PLL, we used a high-performance encoder (Heidenhain's EQI1131). Fig. 17 shows the experimental system for testing the proposed idea. We connected Heidenhain's absolute optical encoder EQI1131 and the AMPME encoder via coupling, then controlled the motor to rotate the position as desired. Absolute optical encoders are suitable for the comparative verification of the AMPME because the optical disk is higher than the pole-pitch precision of a MPM. The position data of EQI1131 is transferred to the encoder signal processing system, and then the data is compared. Through the MCU emulator, AMPME data and EQI1131 data are transmitted to the PC. To compare these transmitted data sets, we performed frequency analyses using Matlab to find the calculated error between these two data sets. We then repeated the above steps for each algorithm to compare the two PLL algorithms (CPLL, AADPLL) and the linear compensator.

Fig. 18 shows the error results of constant frequency case. The errors of CPLL and AADPLL indicated that the high-frequency error was eliminated. However,

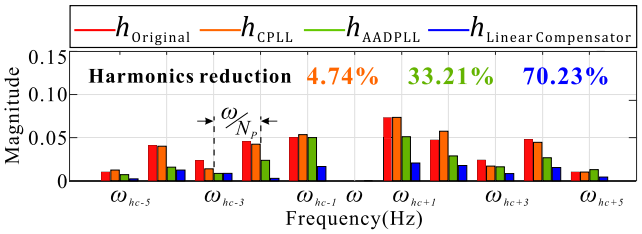

FIGURE 20. Comparison of harmonic components.

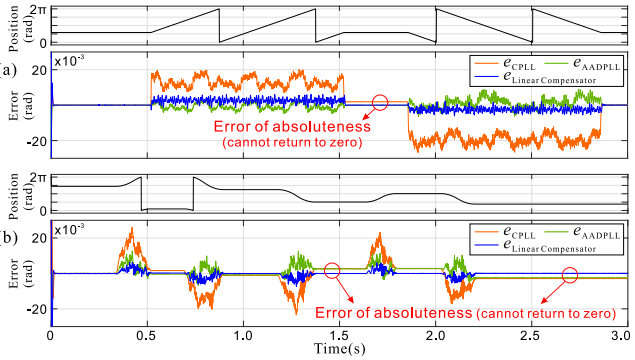

FIGURE 21. Results for inconstant frequency signals. (a) is results of the frequency step case and (b) is results of the accelerating case.

the low-frequency error was still present. On the other hand, the linear compensators are shown to reduce the low-frequency errors. As a result, we confirmed that the error improved: CPLL case, $\pm 6.6\times10^{-10}(rad)$; AADPLL case, $\pm 3.8\times10^{-10}(rad)$; and linear-compensator case, $\pm 2.5\times10^{-10}(rad)$. Then, we analyzed the FFT frequency and confirmed the harmonic rejection. Fig. 19 and 20 show the characteristics of the three compensation methods and the original error without compensation. CPLL and ADDPLL reduced the harmonic components outside of the main-frequency band. However, these methods did not affect the harmonic components of the main-frequency band. On the other hand, the proposed linear compensator slightly reduced the harmonic components outside of the main-frequency band, and greatly reduced the harmonic component of the main-frequency band. Fig. 20 demonstrates that linear compensators are more effective at reducing harmonic components in the main frequency band than the CPLL and ADDPLL approaches. Compared with the harmonic components of the AMPME in the main-frequency band before compensation, CPLL reduced the harmonic component error by 5% and ADDPLL by 33%. The linear compensator reduced the harmonic component error by 70%, making it about twice as effective as the PLL.

This paper presents an analysis of changes in the performance of non constant frequency case. We measured AMPME's position errors by inputting step frequency step case and accelerating case to the motor. Fig. 21 shows the resulting DC error. For the step frequency input, the DC error of the CPLL increased, and an absolute error occurred even when the rotation stopped. The AADPLL approach never caused an error when the rotation stopped, but the DC error

TABLE 1. Comparison of techniques.

Techniques	CPLL	AADPLL [16]	Linear Compensator
Accuracy $(rad) \times 10^{-3}$ at constant frequency case	± 6.6	± 3.8	± 2.5
Accuracy $(rad) \times 10^{-3}$ at accelerating time case	± 26.0	± 9.7	± 6.0
Accuracy $(rad) \times 10^{-3}$ at stopped case	± 2.4	± 1.3	± 0.2
Harmonic Reduction	5%	33%	70%
Noise Reduction (random noise)	✓	✓	✓
Calibration (amplitudes and offsets)	-	-	✓
Phase compensate (for pole-pitch difference)	-	-	✓
Processing time (Cortex-M4, $168MHz$)	$8us$	$25us$	$10us$

increased. And for the accelerating case, both the CPLL and AADPLL had an absolute position error when the rotation stopped. On the other hand, the linear compensator had less DC error, and no absolute error occurred. This means that unlike CPLL and AADPLL, the linear compensator is not sensitive to frequency changes. Table 1 shows the comparison between constant frequency case, accelerating case, and absolute position error after rotation stopped. Also, the calculation time of the processor used in the AMPME was measured and compared. This took 8*us* for CPLL, 25*us* for AADPLL, and 10*us* for the linear compensator. In terms of frequency change, the linear compensator is the most economical.

VI. CONCLUSION

In this paper, we have described the factors that lessen the absolute position accuracy of an AMPME, and proposed a linear compensator to overcome these issues. This method was more effective than the PLL techniques at compensating for the signal and did not degrade the accuracy of absolute position. In particular, we were able to effectively compensate the pole-pitch difference of the MPM. To demonstrate this, we developed an AMPME and a Cortex-M4F MCU-based signal processing system. Then, we compared the absolute position results with other high-performance encoders.

Table 1 shows the comparison of the PLL methods and linear compensator. The proposed linear compensator can effectively eliminate harmonic component errors due to the hardware problems present in the AMPME. In addition, CPLL cannot remove DC errors; AADPLL overcomes this issue. However, the method is sensitive to frequency changes, cohere a stop after acceleration, decreases the absolute position accuracy. This is well explained in the CPLL error formula (6). When a frequency step or frequency ramp input is applied to (6), the error does not converge to 0 as shown in formulas (8) and (9). ADDPLL is a higher-order algorithm than CPLL. Therefore, it is possible to converge the error to 0 in a simple frequency step or frequency ramp input, but it is not sufficient in situations where the frequency changes non-linearly. Graph (b) in Fig. 21 is the motion in which the frequency of the motor increases or decreases non-linearly. An additional reason for this problem; it is basic that use the computation of the PLL algorithm is time-dependent. In PLL characteristics, the higher is the sampling (i.e., the higher is the bandwidth of the sensor), the better will be the results. If the system often checks and adjusts the results, it will obtain a more accurate result.

We demonstrated that the proposed linear compensator's signal compensation has higher performance than the PLL method and can preserve its absolute position without regard to frequency change of the shaft. In other words, the linear compensator is not affected by changes in time or frequency. This robustness is the most important factor for an absolute encoder. Also, the linear compensator is very simple compared to other methods. A small lookup table ($N_p \times 18$ parameters) are used in the static calibrator.

CPLL has achieved adequate results when replacing the arctangent function (to calculate phase). This is an important idea in the signal processing technology of encoder. AADPLL improves the performance of the PLL considerably, but it requires a lot of computation time. This means that the high-order of the PLL algorithm, the longer will be required time (because calculation is more complex).

On the other hand, the proposed linear compensator can calculate the phase by replacing the arctangent or PLL, which greatly reduces the computation time and does not decrease performance. This is possible because the encoder signal processor completed the necessary calculations in pre-calibration. Therefore, the proposed linear compensator is relatively simple, fast, and economical compared to the PLL method.

The biggest drawback of the PLL-based algorithm is that the absolute position cannot be accurately compensated if the previous position information is insufficient. However, the proposed linear compensator does not this problem. Since parameter values obtained during encoder calibration in advance are stored (registered), there is no need to maintain the encoder operation by always maintaining power. This is very effective for applications involving absolute magnetic encoders in industrial sites.

REFERENCES

[1] K. Miyashita, T. Takahashi, and M. Yamanaka, "Features of a magnetic rotary encoder," *IEEE Trans. Magn.*, vol. TMAG-23, no. 5, pp. 2182–2184, Sep. 1987.
[2] Y.-J. Luo, E.-T. Hwang, and S.-M. Huang, "Multi-pole magnetization of high resolution magnetic encoder," in *Proc. Elect./Electron. Insul. Conf.*, Chicago, IL, USA, Oct. 1993, pp. 237–242.
[3] S. Masayuki, Y. Hirasawa, and S. Ozeki, "Absolute encoder device and motor," U.S. Patent 9 379 598, Jun. 28, 2016.
[4] T. Takahashi, T. Nakajima, H. Hakamata, I. Uemoto, S. Miyazaki, T. Harano, and T. Oda, "Magnetic encoder device and rotation detection device," U.S. Patent 9 976 874, May 22, 2018.
[5] C. S. Chou, T. W. Shin, H. F. Jen, and H. S. Ming, "Disk type of absolute-position magnetic encoder for rotary devices," U.S. Patent 5 757 180, May 26, 1998.
[6] S. Hao, Y. Liu, and M. Hao, "Study on a novel absolute magnetic encoder," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Bangkok, Thailand, Feb. 2009, pp. 1773–1776.

[7] S.-T. Wu and Z.-L. Wang, "Equilateral measurement of rotational positions with magnetic encoders," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 10, pp. 2360–2368, Oct. 2016.
[8] C.-S. Wu and A.-Y. Wu, "Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 6, pp. 548–561, Jun. 2001.
[9] D. Zheng, S. Zhang, Y. Zhang, and C. Fan, "Application of CORDIC in capacitive rotary encoder signal demodulation," in *Proc. 8th IEEE Int. Symp. Instrum. Control Technol. (ISICT)*, London, U.K., Jul. 2012, pp. 61–65.
[10] T. N.-C. Tran, H. X. Nguyen, J. W. Park, and J. W. Jeon, "Improvement of the accuracy of absolute magnetic encoders based on automatic calibration and the fuzzy phase-locked-loop," in *Proc. IECON-43rd Annu. Conf. IEEE Ind. Electron. Soc.*, Beijing, China, Oct. 2017, pp. 3310–3315.
[11] H. X. Nguyen, T. N.-C. Tran, J. W. Park, and J. W. Jeon, "Auto-calibration and noise reduction for the sinusoidal signals of magnetic encoders," in *Proc. IECON-43rd Annu. Conf. IEEE Ind. Electron. Soc.*, Beijing, China, Oct. 2017, pp. 3286–3291.
[12] S.-Y. Jung and K. Nam, "PMSM control based on edge-field Hall sensor signals through ANF-PLL processing," *IEEE Trans. Ind. Electron.*, vol. 58, no. 11, pp. 5121–5129, Nov. 2011.
[13] T. Emura and L. Wang, "A high-resolution interpolator for incremental encoders based on the quadrature PLL method," *IEEE Trans. Ind. Electron.*, vol. 47, no. 1, pp. 84–90, Feb. 2000.
[14] A. Bellini and S. Bifaretti, "A digital filter for speed noise reduction in drives using an electromagnetic resolver," *Math. Comput. Simul.*, vol. 71, nos. 4–6, pp. 476–486, Jun. 2006.
[15] M. Benammar, L. Ben-brahim, M. A. Alhamadi, and M. Al-Naemi, "A novel method for estimating the angle from analog co-sinusoidal quadrature signals," *Sens. Actuators A, Phys.*, vol. 142, no. 1, pp. 225–231, Mar. 2008.
[16] H. V. Hoang and J. W. Jeon, "An efficient approach to correct the signals and generate high-resolution quadrature pulses for magnetic encoders," *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3634–3646, Aug. 2011.
[17] M. Ciobotaru, R. Teodorescu, and V. G. Agelidis, "Offset rejection for PLL based synchronization in grid-connected converters," in *Proc. 23rd Annu. IEEE Appl. Power Electron. Conf. Expo.*, Austin, TX, USA, Feb. 2008, pp. 1611–1617.
[18] G. Ye, H. Liu, Y. Wang, B. Lei, Y. Shi, L. Yin, and B. Lu, "Ratiometric-Linearization-Based high-precision electronic interpolator for sinusoidal optical encoders," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 8224–8231, Oct. 2018.
[19] J. W. Park, H. X. Nguyen, T. N.-C. Tran, and J. W. Jeon, "Improve efficiency multi-turn magnetic encoder that uses gear system," in *Proc. 17th Int. Conf. Control, Autom. Syst. (ICCAS)*, Jeju, South Korea, Oct. 2017, pp. 318–324.
[20] A. Houda, "Multi-turn absolute rotation angle detection device and method of detecting absolute rotation angle," U.S. Patent 9 528 855, Dec. 27, 2016.

JAE WAN PARK received the B.S. degree in mechatronics engineering from Korea Polytechnic University, Siheung, South Korea, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea. His research interests include embedded systems, automation systems, signal processing, and real-time applications.

HA XUAN NGUYEN received the B.S degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2015, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2020.

He is currently a Postdoctoral Researcher with the College of Information and Computer Engineering, Sungkyunkwan University. His research interests include signal processing, motion control, robotics, and embedded systems.

THUONG NGOC-CONG TRAN received the B.S degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2015, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2020.

He is currently a Postdoctoral Researcher with the College of Information and Computer Engineering, Sungkyunkwan University. His research interests include signal processing, motion control, robotics, and robot vision.

JAE WOOK JEON (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1990.

From 1990 to 1994, he was a Senior Researcher with Samsung Electronics, Suwon, South Korea. Since 1994, he has been with Sungkyunkwan University, Suwon, where he was first an Assistant Professor with the School of Electrical and Computer Engineering and is currently a Professor with the School of Information and Communication Engineering. His research interests include robotics, embedded systems, and factory automation.