Data Age Error Compensation for Nonconstant Velocity Metrology

Chen Wang, *Graduate Student Member, IEEE*, and Jonathan D. Ellis, *Member, IEEE*

***Abstract*— Positioning calibration under dynamic conditions is becoming increasingly of interest for high precision fields, such as semiconductor lithography and additive manufacturing. Because heterodyne interferometry has high dynamic range and direct traceability to the meter, it is widely used in stage position calibration. When processing measurement and reference signals of heterodyne interferometry, the signal processing algorithm and optical receiver introduce time delay and phase shift to the measurement data, which leads the measurement data cannot precisely represent the current position of the stage. That error is called data age error and depends on the velocity of the stage, which will become significant when stage's velocity is nonconstant in dynamic calibrations. This paper presents a comprehensive data age error model and a compensation method by tracking the derivatives of the phase and solving for the data age error in a field programmable gate array (FPGA) in real time. The FPGA hardware-in-the-loop simulation shows that this method can significantly decrease the displacement error from ±600 to ±0.8 nm (fixed delay time) in dynamic cases, and it will still keep subnanometer resolution for quasi-static calibrations.**

***Index Terms*— Data age, displacement measurement, error compensation, field programmable gate arrays (FPGAs), interferometry, phase measurement.**

I. Introduction

DISPLACEMENT measuring interferometry is an optical dimensional metrology technique, which has high bandwidth, high dynamic range, and direct traceability to meter [1]. It is widely used displacement metrology, position control, position sensor calibration, and stage calibration. A typical stage calibration usually occurs when the stage is held statically, and then measures a series of discrete positions, or the stage moves slowly in a quasi-static configuration. This calibration processing only measures the static performance and the position accuracy of the stage, but independently of the dynamic performance of the stage. However, the quasi-static motion profiles often do not reflect the ultimate intended use of the stage.

Manuscript received April 11, 2016; revised June 11, 2016; accepted July 10, 2016. Date of publication August 16, 2016; date of current version October 7, 2016. This work was supported in part by the U.S. Department of Commerce, National Institute of Standards and Technology under Award 70NANB14H262 and in part by the National Science Foundation under Award CMMI:1265824. The Associate Editor coordinating the review process was Dr. George Xiao.

C. Wang is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: chen.wang@rochester.edu).

J. D. Ellis is with the Department of Mechanical Engineering, The Institute of Optics, University of Rochester, Rochester, NY 14627 USA (e-mail: j.d.ellis@rochester.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2016.2597998

In practice, the intended use of many stages is traveling repetitively along a preprogrammed path with rapidly changed velocity. Hence, its dynamic performance and position accuracy must be well known. However, existing machine tool standards [2]–[4] only define explicit procedures for quasi-static calibrations. Recently, there is an initiative to establish a standard for performing dynamic stage calibrations to complement existing standards for quasi-static calibrations [5], [6]. In this paper, we investigate how the signal processing for heterodyne interferometry systems is affected by the nonconstant velocity profiles found when dynamically measuring the position of a stage.

Interferometer architectures can vary depending on the desired target geometry and intended application [7]. In this paper, our study is based on the heterodyne interferometer, whose metrology principle is: the target position changes are recorded as a Doppler frequency shift in the optical frequency, which manifests as a measured phase change. The target displacement is determined by the measured phase change

$$x = \frac{\lambda \phi}{2\pi N n} \tag{1}$$

where x is the displacement of the target, ϕ is the phase difference between the reference and measurement signals, N is the interferometer fold factor (two for the interferometer used in this paper), n is the refractive index along the optical path difference, and λ is the nominal wavelength of the laser light.

The timeliness of the measurement data is important for some measurement, calibration, and control applications. The measurement data are expected to be available right after the measurement occurs; thus, the data can precisely represent the current position and motion status of the target. Since the displacement information measured by heterodyne interferometry is encoded in the phase difference between two laser beams (1), to extract the displacement information, an electrical phase measurement instrument is used, which is also called phasemeter (see Fig. 1) [8], [9]. However, the analog and digital devices, and the signal processing algorithm introduce inevitable phase shift and time delay to the measurement data. The phase shift and time delay impact the timeliness of the measurement data.

The data age error is the measurement error due to the time difference between the occurrence of a measurement and the time when the measured data are available to the user [10]. Fig. 2 shows the comparison between actual displacement and measured displacement. In terms of time, it is a time delay

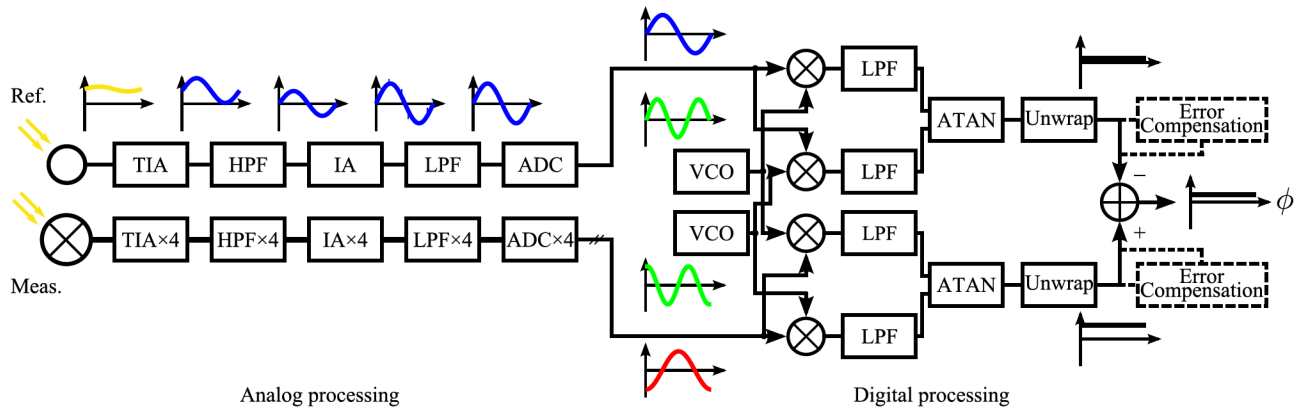


Fig. 1. Schematic of the heterodyne interferometry phase measurement system. It consists of two parts, an analog processing part and a digital processing part. The analog processing system employs photodiodes, amplifiers, and filter, to process the analog signal, and improve the quality of the signal. The digital processing system used an SBDFT technique to demodulate phase difference and was implemented in an FPGA in this design [8], [9].

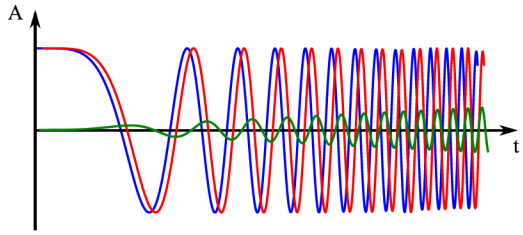


Fig. 2. Comparison between the actual displacement (blue curve) and the measured displacement (red curve) of a sinusoidal motion. Due to the transmission and processing, there is a time delay between the measured displacement and the actual displacement, which will be a measurement error (green curve).

between two signals, but in terms of phase (displacement), it is a measurement error. It should be noted that this error is not limited to interferometry as linescales would exhibit a similar behavior as a function of velocity for their phase measurements as well.

There has been limited reporting of the data age error in the literature for displacement (phase) measurement. Some literature has discussed the inaccuracies of the phase measurement [11], [12], and presented a similar concept to data age error, but none provide methods for compensating this error. Djokic and So [13] also found that the phase shift (one source of data age error) introduced by filters will influence the phase measurement, but they state that it does not cause appreciable change. However, in dynamic, precise, and high-frequency measurements, more applications find that the timeliness of the measurement determines the accuracy, such as in satellite navigation systems [14] and displacement measuring interferometry [1], [9], [15], [16]. For displacement measuring interferometry, this error and potential compensation methods have only been discussed in the context of constant velocity [1], [16]–[20].

In this paper, we investigate the sources of data age error, model the effects of this error, and propose methods for compensating it for heterodyne interferometry applications, especially for nonconstant velocity motion measurements. We present compensation methods for both fixed and variable delay, and discuss the limitation of each method and perform simulations to validate the proposed methods.

II. Heterodyne Interferometry Signal Processing

The displacement of the target can be determined by measuring the phase difference between two optical signals, one, a nominally fixed optical reference signal, and two, an optical signal with the displacement encoded as continuous changes in phase from a Doppler shifting target. For heterodyne systems, these signals essentially amount to an instantaneous frequency modulation between the reference and measurement signals. To obtain the phase difference, the two optical signals are detected and converted to voltage level representations of interference amplitude, typically called the measurement signal and reference signal

$$u_m(t) = U_m \cos(2\pi f_s t + \phi_m) + u_{\text{dc}} + u_{\text{noise}} \tag{2}$$

and

$$u_r(t) = U_r \cos(2\pi f_s t + \phi_r) + u_{\text{dc}} + u_{\text{noise}} \tag{3}$$

where U_m and U_r are the amplitudes of the measurement and reference signals, f_s is the nominal heterodyne frequency of the laser source, ϕ_m and ϕ_r are the phases of the two signals, and u_{dc} and u_{noise} are the dc offset and noise, which may differ in the two signals. The photodiodes and transimpedance amplifiers (see Fig. 1) are for the detection and conversion work, and then high-pass filter, inverting amplifier, and low-pass filter (LPF) attenuate the high-frequency noise u_{noise}, scale the amplitudes u_m, u_r, and remove dc offsets u_{dc}, respectively.

After performing the analog processing, analog-to-digital converters convert the treated measurement and reference signals to digital representations for further digital processing. The general phase measurement techniques are a phase-locked loop, time interval analysis, and single-bin discrete Fourier transform (SBDFT) [7], [8], [21]. In this paper, the SBDFT algorithm is used, whose schematic is shown as the digital processing part of Fig. 1. Voltage controlled oscillators (VCOs) are used to generate a pair of in-phase $u_{v,i}$ and quadrature $u_{v,q}$ signals nominally at a frequency of f_s

$$u_{v,i}(t) = \cos(2\pi f_s t + \phi_v) \tag{4}$$

and

$$u_{v,q}(t) = \sin(2\pi f_s t + \phi_v). \tag{5}$$

Then, the in-phase and quadrature signals are multiplied by the digital measurement signal (2) and reference signal (3) (without the dc offset and noise), respectively. Based on the trigonometric product-to-sum identity, the products equal to a sum of one high-frequency ($4\pi f_s$) component and one low-frequency or quasi-dc ($\phi_{m/r} - \phi_v$) component.

The following LPFs block the high-frequency terms. Thus, the products for the measurement signal can be simplified as

$$u_{m,i} = \frac{1}{2}\cos(\phi_m - \phi_v) \tag{6}$$

and

$$u_{m,q} = -\frac{1}{2}\sin(\phi_m - \phi_v). \tag{7}$$

An arctangent operation is then used to determine the phase difference between the measurement signal and the VCO signals $\phi_m - \phi_v$. A similar process also computes $\phi_r - \phi_v$ for reference signal.

In the end, a subtraction operation is applied to extract the difference between two output phases. The common phases from the VCO signals are canceled, resulting in the desired phase difference $\phi_m - \phi_r$ between the measurement and reference signals.

III. Data Age Error

Typically, there are two sources that can lead data age error. One is the time delay, which is introduced by signal processing time, path length [1], and usually is fixed. Another is the phase shift introduced by analog and digital devices. They add an extra phase to the passing signal, depending on the instantaneous frequency of the signal [9].

A. Time Delay

1) Signal Processing Time: To extract the phase difference between measurement and reference signals, a series of operations must be performed [8], [9], e.g., multiplication, arctangent, and unwrap. Executing these operations costs time, so the result cannot be available immediately. It leads a time delay between input data and corresponding output data [18].

For instance, if using a digital signal processor to process the input signals, the filters are usually implemented as loops, which may take tens or hundreds of clock cycles to execute. If using a field programmable gate array (FPGA), it has good parallelism and dedicated hardware for each operation, but still needs several clock cycles to execute complex operations. In our design, the atan2 operation needs six clock cycles and the unwrap operation needs two clock cycles to generate output.

Once the signal processing algorithm is determined, the number of clock cycles is usually fixed. The signal processing time cannot be eliminated, but there are some ways to shorten it. For example, by increasing the clock frequency, the entire execution time could decrease, or using a more compact design to reduce the number of clock cycles. One can even shift the data back by off-line processing, if the data are headed for recording purposes. However, for online, real-time processing, the processing time delay cannot be eliminated completely.

2) Path Lengths: Parallel signals should be processed simultaneously, or experience the same time delay. For instance, the spatial angle measurement can be a further processing of multiaxis displacement measurements. To achieve high angle measurement accuracy, the different data age errors among multiaxis measurements should be equalizd [1]. Similarly, in differential wavefront sensing [22]–[25], the four signals from quadrant photodiode are also required to have the same delay.

Due to the design and configuration, the signals may not pass the same length path, such as different electrical cables, the path on printed circuit board [15], and even fiber optic cable length [26], which leads the signals to experience different delays [17]. The unequal delays introduce errors when a value is produced by those signals together.

3) Time Delay Model: Assuming the time delay between the input signal and the corresponding output signal is τ, the relationship between actual phase ϕ and measured phase ϕ_d with delay is

$$\phi_d(t) = \phi(t - \tau). \tag{8}$$

B. Phase Shift

The electrical devices, such as filters, amplifiers, and photodiodes, treat (attenuate, amplify, or convert) the signal with respect to their frequency. Besides manipulating the amplitude of the signals, it also adds a phase shift to the passing signal. The relationship between the phase of the output and input signals is the phase response, which is usually varied with changing frequency.

1) Digital Filters: Digital filters are used to remove high-frequency components for demodulating phase difference between measurement and reference signals [8]. When the desired signal passes through the filters, the filters add a phase shift to the output signal with respect to their phase response. This phase shift causes the phase measurement inaccuracy in dynamic conditions [9].

Two types of digital filters are commonly used in the phase demodulation, finite impulse response (FIR), and infinite impulse response (IIR) filters. FIR has a linear phase response, but a nonflat passband, and costs more hardware resources for FPGAs. IIR has a nonlinear phase response, but a flat passband, and is straightforward for design. The phase responses of an FIR and IIR filter are shown in Fig. 3.

The phase shift can be regarded as a type of time delay. The linear phase response leads constant or fixed time delay, and the nonlinear phase response leads nonconstant or variable time delay. They need different compensation strategies, which will be illustrated in Section IV.

To eliminate the phase shift, a zero-phase response is expected, which means the phase of the signal does not change when passing the device. However, the zero-phase filter is a noncausal system, whose current output depends on the future input. It is impossible for it to process the streaming data in real time.

2) Optical Receiver: The optical receiver is the analog part of the phasemeter, which consists of photodiodes, amplifiers,

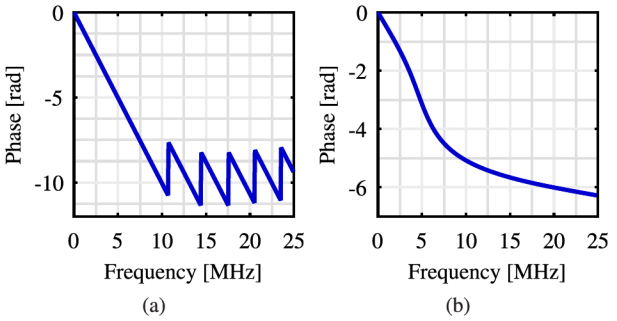


Fig. 3. Phase responses of (a) FIR and (b) IIR. The cutoff frequency of both the filters is 5 MHz. The FIR filter is 17 tap, and IIR filter is the fourth order. The phase response of FIR is linear in the passband, while that of IIR is nonlinear. However, FIR introduces more phase shift than IIR at the same frequency.

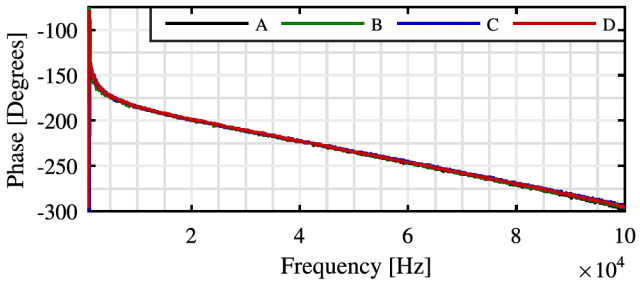


Fig. 4. Phase response of an optical receiver. The optical receiver is designed as a part of our phasemeter system (Fig. 1). It uses a quadrant photodiode, so it has four parallel channels. The phase responses of these four parallel channels look identical and linear globally, and however, they still have a small difference and nonlinearity in their working band.

and filters (see Fig. 1) [8], [19]. It converts the optical signal to an electrical signal via analog signal processing.

Each of those analog devices has its own working frequency band, and in these bands, the phase responses usually are nonlinear. The phase response of the optical receiver is the superposition of those devices, which is nonlinear as well [19]. Fig. 4 shows the phase response of an optical receiver with a quadrant photodiode.

3) Phase Shift Model: The phase response is the phase shift with respect to the frequency of signal. When a signal passes a filter, the output signal has a phase shift compared with the input signal. This phase shift $\Delta\phi(\omega)$ can cause time delay (phase delay)

$$\tau_\phi = -\frac{\Delta\phi(\omega)}{\omega}, \tag{9}$$

where τ_ϕ is the phase delay at frequency ω, which represents the slope of the phase response at frequency ω. So the relationship between actual phase ϕ and measured phase ϕ_d can be written as

$$\phi_d(t) = \phi(t) + \Delta\phi = \omega t - \omega\tau_\phi = \phi(t - \tau_\phi) \tag{10}$$

which has the same expression as (8). However, the phase delay τ_ϕ is not necessarily a constant value, depending on the linearity of the phase response of devices [1]. The FIR filter has a linear phase response, so the phase delay introduced by FIR is a constant value. The IIR filter and analog devices have a nonlinear phase response, so the phase delay is nonconstant and varied with the frequency.

IV. Compensation Algorithm

Several previous studies have investigated the sources of the data age error, and presented different ways to estimate data age error from constant [18] and nonconstant [19] time delay. Their methods assume that the measurement error caused by time delay is linearly related to the velocity (Doppler frequency, the first derivative of phase). By continuously tracking the velocity of the target, the displacement measurement error can be calculated by the multiplication of velocity and time delay [1], [7], [10]

$$e_{DA} = \tau v \tag{11}$$

where τ is the equivalent time delay and v is the velocity of the target. For example, if the measurement system has 1 μs time delay, and the target is moving at 1 m/s, the displacement measurement error is 1 μm.

To compensate the constant time delay, the time delay τ is obtained in advance by calibration or theoretical derivation. A module tracks the Doppler frequency f_D (equivalent to velocity) by calculating the first derivative of the raw phase measurement in real time, and then the actual phase ϕ can be calculated by [1], [15], [18], [20]

$$\phi(t) = \phi_d(t) - \Delta\phi = \phi_d(t) + 2\pi f_D \tau. \tag{12}$$

To compensate the nonconstant time delay, the time delay at each Doppler frequency is measured in advance. Since the time delay is varied with the frequency, the error cannot be compensated as (12). Usually, it stores a finite number of time delay $\tau(f_D)$ or phase error $\Delta\phi(f_D)$ values into a lookup table (LUT), and uses the instantaneous Doppler frequency f_D as the address to access to corresponding time delay or phase error [9], [18], [19]. Then, the actual phase can be calculated by

$$\phi(t) = \phi_d(t) - \Delta\phi(f_D). \tag{13}$$

This method will introduce quantitative error, since LUT cannot store a phase error for every potential frequency. An alternative method is using polynomial curve fitting to find the approximate expression of the phase error $\Delta\phi(f_D)$ [20], and then calculate the phase error for instantaneous Doppler frequency

$$\phi(t) = \phi_d(t) - \sum_{i=0}^{n} p_i \cdot {f_D}^i \tag{14}$$

where p_i is the coefficient of polynomial fitting. However, the polynomial calculation uses significantly more hardware resource.

Those previous studies all focuses on constant velocity motion, and their methods to compensate the data age error only work well for constant velocity motion. However, more and more the nonconstant velocity motion scenarios appear in measurement, calibration, and control applications. When measuring a rapidly changed velocity motion, the above models cannot represent the data age error precisely. For example,

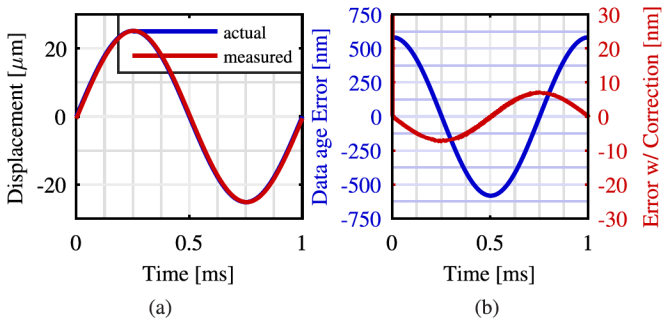


Fig. 5. Measurement of a sinusoidal motion. (a) Comparison between actual and measured displacement. The maximum velocity is 158 mm/s (Doppler frequency 500 kHz), and the oscillating frequency is 1 kHz. There is a very small time delay between the two signals. (b) Time delay leads a ±600 nm error (blue curve). Using the old method to compensate the error, there is still ±6 nm error left (red curve).

in Fig. 2, the velocity of the sinusoidal motion is nonconstant. According to (11) and (12), the error should be zero when the velocity is zero. However, top points (at zero velocity) of blue and red curves are not overlapping, which means the actual error is not zero. Fig. 5 gives another example, which shows the old method cannot completely compensate the data age error for nonconstant velocity motion.

Since the old method is not effective for nonconstant velocity motion, we present an upgraded model for the data age error, and a method to compensate it for all possible motions. From the simulations, we find that the data age error is related to not only the velocity of the target, but also the acceleration and even higher order derivatives of the displacement (phase). Thus, our proposed method is to add more higher order terms to represent the data age error.

If we express actual phase in terms of the measured phase and its Taylor expansion, the actual phase is given by

$$\phi(t) = \phi_d(t+\tau) = \phi_d(t) + \sum_{n=1}^{\infty} \frac{{\phi_d}^{(n)}(t)}{n!}\tau^n \qquad (15)$$

where $n!$ denotes the factorial of n and ${\phi_d}^{(n)}(t)$ is the nth derivative of measured phase ϕ_d. The first derivative is velocity; the second derivative is acceleration. Compared with (15), we can find that (12) is a simplified version of (15). Since (15) includes more terms related to the status of motion, it is more comprehensive to represent data age error for any motions.

In this paper, we present a comprehensive model for compensating the data age error for nonconstant velocity motion, and implement it and integrate it with our FPGA-based phasemeter to achieve online and real-time compensation. It can increase the measurement precision in the dynamic conditions, and also keep the real-time performance without adding extra computation time.

V. Implementation

Conceptually, the solution to compensate the data age error is to subtract the each order of phase error from the measurement result. The procedure of implementing the compensation of the data age error includes the following steps.

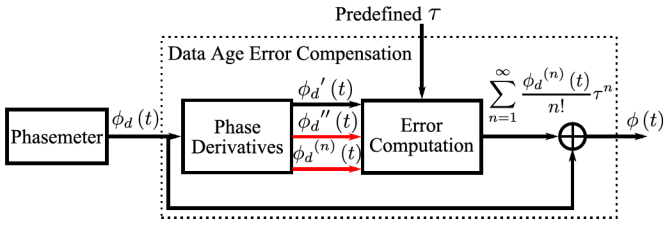


Fig. 6. Schematic of the DAEC module. It consists of two parts, the instantaneous phase derivatives tracking and the error computation part. First, each order of derivative of the raw phase ϕ_d is extracted. Then, the corresponding phase shift errors $\phi_d^{(n)}(t)\,\tau^n/n!$ is solved by the instantaneous derivatives $\phi_d^{(n)}(t)$ and predefined τ. Last, each order of error and the raw phase sum up to get actual phase ϕ without data age error.

1) Determine the equivalent time delay τ of the system before the measurement.
2) Calculate the first-, second-, or even higher order derivatives of raw phase in real time during measurement.
3) Compute the data age error using equivalent time delay and phase derivatives in real time.
4) Subtract the error from the raw phase measurement.

The schematic of overall data age compensation implementation is shown in Fig. 6, which includes an instantaneous phase derivatives tracking part and an error computation part. The difference between the current and previous implementation is that the current one also calculates the second- or higher order derivatives of the phase (highlighted), which could represent the motion more comprehensively. The module processes the raw phases $\phi_d(t)$, and generates compensated phases $\phi(t)$ in real time.

A. Time Delay Determination

1) Theoretical Derivation: The fixed time delay can be derived through theory. Our signal processing module is implemented in an FPGA. The time delay from signal processing can be determined by counting the number of registers or pipelines between input and output. The signal is going to delay one clock cycle at each register on the signal path. Basic operations, such as addition and multiplication, can finish within a clock cycle. Complex operations need multiple clock cycles to generate the result.

As mentioned before, FIR filters have linear phase response, so the phase delay (time delay) it introduced is a constant value. The phase delay can be determined by the equation

$$\tau_\phi = \frac{N-1}{2} \cdot F_s \qquad (16)$$

where N is the taps or stages the FIR filter has and F_s is the sampling rate of the system. In this design, we use a 31-tap FIR filter, so it is going to introduce 15 clock cycles delay.

For some optical and electrical components and devices, their manuals may provide the time delay through factory calibration [18].

2) Measurement: In some cases, the time delay cannot be obtained, or not known by a theoretical derivation. Thus, the actual delay must be measured.

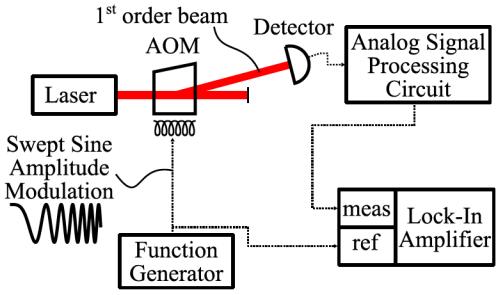


Fig. 7. Experimental setup for measuring the analog system's phase response. The input laser is diffracted in the AOM that is amplitude modulated with a swept sine over the desired frequency range. It assumes the 6-ns response time of the AOM imparts little delay in the system. This is the setup to measure the phase response of the optical receiver in Fig. 4.

TABLE I
ERROR CONTRIBUTION AS A FUNCTION OF THE DERIVATIVE ORDER

	1st order	2nd order	3+ orders
max. [nm]	582.4	6.74	0.09
rms [nm]	411.8	4.76	0.05

For an analog system, the actual performance and phase profile usually deviates from nominal parameters, which is caused by variations in components' tolerances and performance, and the capacitive coupling in the board and near-field. The phase response of the photodiode is not provided explicitly by the datasheet. So the phase response must be measured to determine the phase shift.

To measure the phase response of our analog system, a function generator was used to drive an acousto-optic modulator (AOM) to modulate the output power of a laser source and vary the light intensity at the drive frequency. Then, a lock-in amplifier was used to compare the phase difference between the input and output signals of the analog processing system. The experimental setup is shown in Fig. 7 [9].

For a digital IIR filter, its phase response is nonlinear, which leads a nonconstant time delay. Also, fixed-point coefficients and operations cause the phase response to be slightly different from the ideal, floating-point design. Thus, the actual phase response must be measured to make sure that the compensation is accurate. The method to measure the phase response of the digital IIR filters is similar to that of the analog system. The measurement is performed on a PC, simulated in MATLAB/Simulink with the Altera DSP Builder toolbox [9].

B. Phase Derivative Calculation

1) Orders: The full expression of (15) consists of an infinite number of derivatives; however, it is impossible to calculate all the derivatives. The error contribution of each derivative must be identified, and balance should be made between simplicity and accuracy. Table I shows the error contribution of each order derivative for our extreme test case, which is the same motion as in Fig. 5. From Table I, the first-order derivative (velocity) contributes the majority of the data age error; the contributions of the third and higher orders are negligible. If requiring subnanometer accuracy, the second-order derivative (acceleration) must be considered. In our implementation,

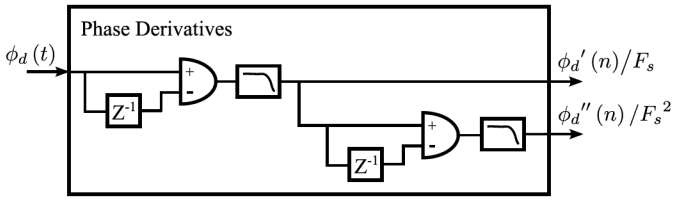


Fig. 8. Implementation of the phase derivatives calculation. Based on (17) and (18), it calculates the first- and second-order derivatives. The additional filters are used to attenuate the noise amplified by the derivative operation. The multiplication with F_s and ${F_s}^2$ is removed to the error computation part.

the first two derivatives are calculated, and the rest of the terms are ignored for simplicity.

2) Implementation: To compensate the phase in real time, the phase derivatives should be calculated in real time, too. Here are some restrictions to calculate or estimate the derivatives [27].

1) The derivatives should be calculated or estimated online and in real time.
2) The raw phase may contain measurement noises, which impacts the derivative calculation.
3) The motion of the target (the shape of phase measurement) is unknown and may be arbitrary. The raw phase is only information can be used.

Our first attempt to calculate the first- and second-order discrete derivatives is based on a finite difference method

$$\phi_d{}'(n) = (\phi_d(n) - \phi_d(n-1)) \cdot F_s \tag{17}$$

$$\phi_d{}''(n) = (\phi_d{}'(n) - \phi_d{}'(n-1)) \cdot F_s \tag{18}$$

where F_s is the sampling rate of the system. This method is straightforward to understand, and the implementation (see Fig. 8) can calculate derivatives instantaneously right after new measurement data are available.

However, it is sensitive to the presence of noise in the signal. The transfer function of the first derivative is

$$H(s) = s. \tag{19}$$

From the equation, we can know it amplifies the noise, especially the high-frequency noise. For the second-order derivative, the signal-to-noise ratio of the result becomes even worse. Numerically calculating derivatives amplifies the noise, which may make the result not useful anymore [28].

There have been some methods to estimate the derivative of the noisy signal in signal processing and system control applications. The basic idea is smoothing the signal first and then calculating its derivative. The common methods include polynomial regression, Tikhonov regularization, smoothing spline, convolution smoothing, and total variation regularization [29]. These methods either need a large number of complex operations or depend on future samples (noncausal systems). So they are impractical to be implemented in an FPGA for online and real time estimating the derivative.

We choose an algebraic derivative estimation method, which uses a finite weighted combination of integrations of a noisy signal to estimate its derivatives for an arbitrarily small amount

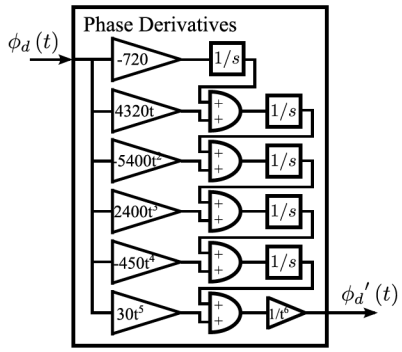


Fig. 9. Schematic of the algebraic derivative estimation method. It only shows the first-order derivative estimation. The second-order derivative estimation is similar to the first order, and shares components with the first-order derivative. The variable t is generated by an internal counter.

of time [27]. It claims to have high robustness properties with regard to influential noises. The principle is introduced in [27] and [30] in detail. A real-time implementation of this method is presented in [31].

If using five orders of integration, the first- and second-order derivatives of raw phase can be estimated as

$$\phi_d'(t) = \begin{cases} \text{arbitrary constant} & 0 \le t < \varepsilon \\ \dfrac{30\phi(t)}{t} + \dfrac{z_1}{t^6} & t > \varepsilon \end{cases} \tag{20}$$

and

$$\phi_d''(t) = \begin{cases} \text{arbitrary constant} & 0 \le t < \varepsilon \\ \dfrac{420\phi(t)}{t^2} + \dfrac{24z_1}{t^7} + \dfrac{z_2}{t^6} & t > \varepsilon \end{cases} \tag{21}$$

where ε is transient time, and

$$z_1' = z_2 - 450t^4\phi(t) \tag{22}$$
$$z_2' = z_3 + 2400t^3\phi(t) \tag{23}$$
$$z_3' = z_4 - 5400t^2\phi(t) \tag{24}$$
$$z_4' = z_5 + 4320t\phi(t) \tag{25}$$
$$z_5' = -720\phi(t). \tag{26}$$

The schematic of this method is shown in Fig. 9. Because the derivative estimations are accurate in only a small amount of time, there is a strategy to reinitialize the computation periodically [27].

3) Performance: As mentioned earlier, the noise in the signal significantly impacts the performance of the derivative calculation. To investigate the influence of the noise, we add different power levels of white noise to the input measurement (2) and reference (3) signals, and check the measurement error. Four different configurations are compared: 1) without using data age error compensation (w/o DAEC); 2) using the first method (17) and (18) but without filtering (1st w/o filter); 3) using the first method with filtering (1st w/ filter); and 4) using the second method (20) and (21) (2nd).

Table II shows the rms of the measurement error when adding white noise to input signals. Here, we are interested in the error introduced by the input noise, and ignore the data age error from the time delay (they are different sources). Without DAEC, after adding 0.2%, 1%, and 5% white noise

TABLE II
MEASUREMENT ERROR RESULTING FROM DIFFERENT LEVELS OF RANDOM WHITE NOISE TO MEASUREMENT AND REFERENCE SIGNALS

Noise Level	S/N [dB]	w/o DAEC	w/ DAEC		
			1st w/o filter	1st w/ filter	2nd
0.2%	54.0	2.38	525.9	4.13	3.21
1%	40.0	3.82	869.8	7.08	5.08
5%	26.0	8.80	1.9e+3	17.6	11.3

* The errors are quoted as rms [nm].

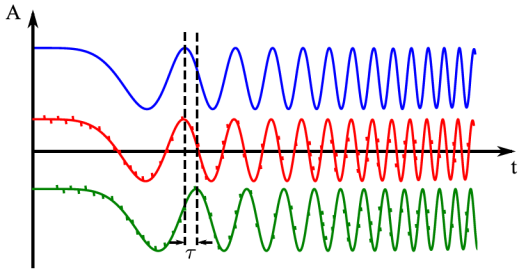


Fig. 10. Comparison between the actual displacement (blue curve), the displacement information provided by the interferometer (red curve), and the measurement result after processing (green curve).

to the input, the corresponding errors appear in the measurement results, whose rms errors are 2.38, 3.82, and 8.8 nm, respectively.

However, using a finite difference method to calculate the derivatives and without filtering (1st w/o filter) significantly amplifies the error. A hundred nanometer level error appears in the measurement result, which significantly worsens the measurement resolution. With proper filtering, the original error is amplified by a factor of 2. Using the algebraic derivative estimation method (2nd), the error is slightly amplified by around 1.3 times, which is better than the first method. The results show that the performance of the last two methods is acceptable when calculating the derivative of a noisy signal.

In practice, these noise or errors on input signals are from upstream optical systems, environmental disturbances, electrical power supply, and so on They can result in a noisy or distorted (such as periodic error [7], [32], [33]) measurement result. The blue curve in Fig. 10 is the displacement what we want to measure, whereas the red curve includes additional noise and distortion. Eventually, the measurement result after processing is the green curve, which has a time delay τ. The goal of this paper is to use the green curve to estimate the red curve precisely (eliminate the time delay), without amplifying the noise too much. As for the error between blue and red curves, effort should be made to optimize the quality of the input signals, which is beyond the scope of this paper but is readily discussed in the literature.

C. Error Computation

The next step to compensate the data age error is solving the error terms $\phi_d'(t)\tau$ and $\phi_d''(t)\tau^2$. The equivalent time delay τ, and phase derivatives are already known. The error terms

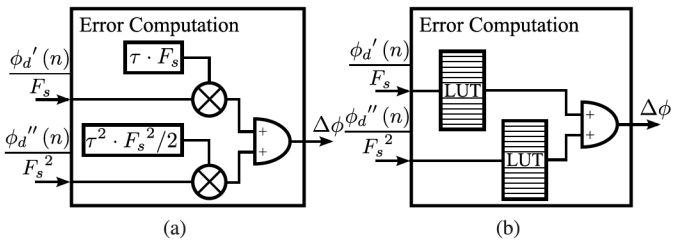


Fig. 11. Implementation of the error computation part. (a) Fixed time delay; it multiplies the derivatives with time delay directly. (b) Variable time delay; it stores precalculated errors in LUTs, and uses derivatives as the address.

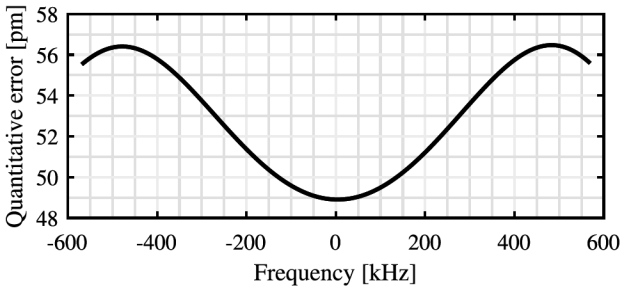


Fig. 12. Quantitative error of LUT. The LUT stores the 8192 points of the first-order error for the Doppler frequency from −780 to 780 kHz, so the frequency resolution is 95 Hz. The quantitative error is around 50 pm, which is acceptable for this design. For the lower quantitative error, a larger LUT volume is expected.

can be solved using (15) directly. However, the equivalent time delay τ actually has two types, fixed time delay τ_f and variable time delay $\tau(f_D)$. For each type of the time delay, we implement an effective solution to compute the error.

For fixed time delay, it is straightforward to multiply the fixed time delay with the derivative directly. Since the fixed time delay τ is known, $\tau^2/2$ also can be known in advance. In the measurement, the instantaneous first and second derivatives can be multiplied by those two constants in real time. The implementation is as shown in Fig. 11(a).

The variable time delay is a function of the derivative of phase. When the target has nonconstant velocity, the time delay changes along with the velocity and acceleration. For every newly calculated derivative, it must find the corresponding time delay first, and then perform the multiplication to get the error. Our solution is to build two LUTs, which store the precalculated first- and second-order error directly, and uses derivative information as an address to access the errors. Once derivatives are determined, and errors are determined as well. By doing this, the additional multipliers are not needed anymore. The implementation is as shown in Fig. 11(b).

However, a potential problem of using LUT is the quantitative error. LUT can only store a limited number of data, which means it has to evenly take samples in the whole range. When the actual derivative is located between two samples, it has to round to the nearest one, which introduces the quantitative error. The volume of the LUT determines the interval between the two samples, and thus determines the quantitative error. In this implementation, we choose an 8192-sample LUT. Fig. 12 shows the quantitative error it introduced.

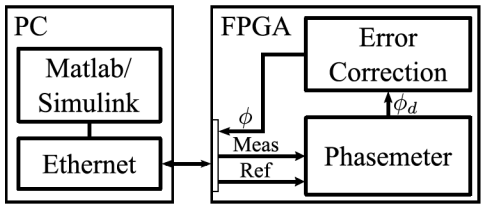


Fig. 13. Setup of HIL simulation. On PC side, stimulus is generated by MATLAB/Simulink, and transmitted to FPGA by Ethernet. On FPGA side, the stimulus is processed by the phasemeter and error correction module, and the result is transmitted back to PC for further processing and analysis.

VI. Verification

To determine the functionality and performance of this DAEC module, a series of verifications were performed. A technique called FPGA hardware-in-the-loop (HIL) was used to verify the functionality and performance of the FPGA design.

Traditional software simulations for FPGA digital signal processing algorithms have a long simulation runtime, and application-specific availability and accuracy. Because the software cannot fully imitate the conditions and environment of the real world, the results may not reflect its real performance in hardware. Unlike software simulation, HIL simulation allows data to be processed in real time by the FPGA hardware rather than by the software. The stimulus data are generated by MATLAB/Simulink, which could be arbitrary or customized, and fed to the FPGA, and then the FPGA computational results are collected by the MATLAB/Simulink for further analysis and display. This approach accelerates simulation time, and also ensures that the algorithm will behave as expected in the real world [34]. The setup of this simulation is shown in Fig. 13.

In the simulations, only the data age error from the digital system is compensated, because the digital system is easy to perform the HIL simulation and its delay time information is easy to obtain. Once the time delay information from other source is determined, it can easily replace or update the predefined time delay in the compensation module.

In this paper, it verifies the overall performance of derivatives tracking and error computation for fixed delay time and variable delay time.

A. Fixed Delay Time

The filters in phase demodulation are implemented as FIR filters, which have linear phase response; thus the phase delay introduced by the filter is fixed. And the digital system has fixed processing time, so the total equivalent time delay is fixed. Several simulations are performed to verify the performance of compensating this fixed time delay. In this design, 37-tap FIR filters and eight-stage cascaded integrator-comb filters are employed, whose cutoff frequency is about 750 kHz. The overall time delay of the system is 3.68 μs.

1) Static: First, a static simulation is performed. When the velocity of the target is zero, according to (15), the data age error should be zero too. However, the precision of the fixed-point operations, high-frequency residuals after filtering, and other sources of noise influences the precision of the

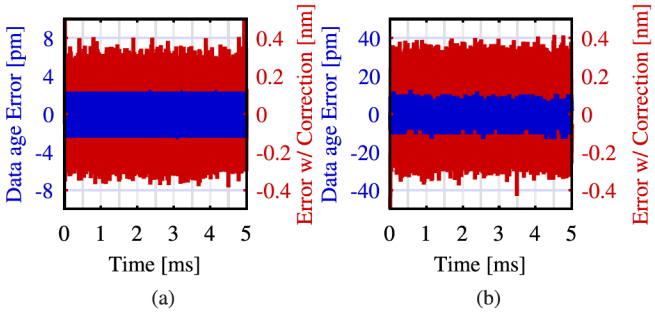


Fig. 14. Data age error (blue curve) and the residual error after compensation (red curve) for (a) static and (b) constant velocity scenarios.

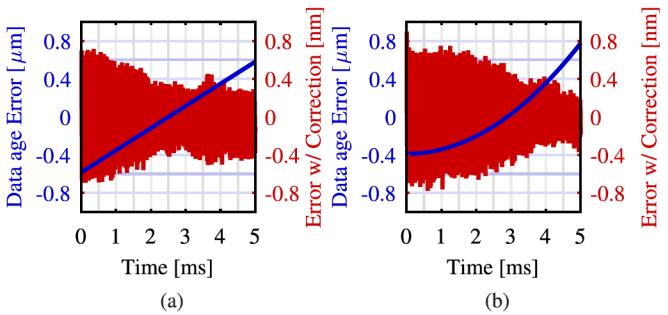


Fig. 15. Data age error (blue curve) and the residual error after compensation (red curve) for (a) linear increasing velocity and (b) linear increasing acceleration scenarios.

phase (displacement) measurement. The new add-on DAEC module should not compensate any phase in the case, but it may introduce some error itself. So the static simulation is to verify how the compensation module influences the static performance of the phase measurement. Fig. 14(a) shows the errors before and after compensation.

When the stage is static, the phasemeter has an error about ±2 pm (blue curve), which determines the displacement resolution. There should be no data age error in a static scenario; however, the compensation module has amplified the error (by derivative operations), which makes the error increase to 0.4 nm (red curve). So the compensation module impacts the displacement measurement a little in this scenario.

2) Constant Velocity: The constant velocity motion is the most common scenario, which all previous studies focus on. The constant velocity motion introduces constant data age error. In fact, it will not impact the displacement measurement, because with constant data age error, the relative distance changes within a certain time interval are still the same.

The 158-mm/s velocity (Doppler frequency 500 kHz) motions have been simulated. Fig. 14(b) shows the errors before and after compensation. Before compensation, the raw error is about ±10 pm (blue curve), which is the residual high-frequency component after filtering. However, after compensation, the error is still amplified.

3) Linear Increasing Velocity: This case is a nonconstant velocity scenario. According to (15), the first derivative is no longer a constant value, and the second derivative is no longer zero, so they introduce a nonconstant data age error. Thus, the measurement cannot represent the current displacement precisely.

A simulation simulates a linear increasing velocity motion from −158 to 158 mm/s (Doppler frequency from −500 to 500 kHz) in 5 ms. Fig. 15(a) shows the errors before and after compensation. The data age error can be up to 0.6 μm at the maximum velocity (blue curve). After compensation, the error is reduced within ±0.8 nm (red curve). Because the filters do not have a consistent magnitude response, the residual error for different velocities is different.

4) Linear Increasing Acceleration: In this scenario, the second derivative starts to introduce nonconstant data age error. A simulation simulates a linear increasing acceleration motion from 0 to 1.58×10^3 m/s^2 in 5 ms. Fig. 15(b) shows the errors before and after compensation. The data age error can be up to 0.8 μm at maximum velocity and maximum acceleration (blue curve). After compensation, the error is reduced within ±0.8 nm (red curve).

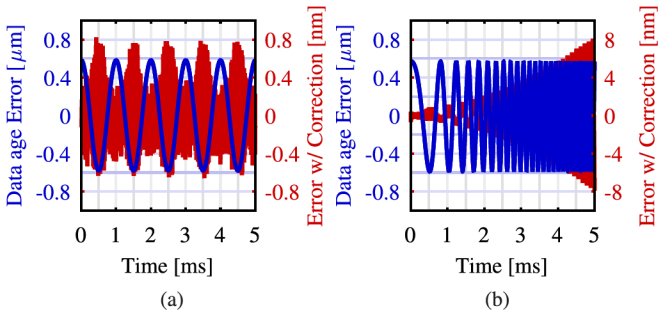


Fig. 16. Data age error (blue curve) and the residual error after compensation (red curve) for (a) sinusoid velocity and (b) sinusoid velocity with varied oscillating frequency scenarios.

5) Sinusoidal Velocity: A sinusoidal velocity is a typical scenario for the movement of a stage in manufacturing processes. Sinusoidal motion profiles are desired for velocity changes, because accelerations and higher order terms (snap, jerk, and so on) are predictable. The difference between this scenario and the scenarios mentioned earlier is that the high-order derivatives in the scenarios mentioned earlier are zero or constant. However, in the sinusoidal velocity scenario, the high-order derivatives start to play a role in producing data age error. Our compensation module is an approximation, which only deals with the first two order errors and ignores the higher order errors. Simulations have been done to explore the performance of this approximation.

A simulation simulates a sinusoidal changed velocity motion. The Doppler frequency of this motion is given by

$$f_D = f_p \cos(2\pi f_o t) \tag{27}$$

where f_p is the peak frequency, which is 500 kHz (158 mm/s) here and f_o is the frequency of velocity oscillating, which is 1 kHz here. Fig. 16(a) shows the errors before and after compensation. The data age error is also oscillating, and the maximum error is about 0.6 μm (blue curve). After the compensation, the residual error keeps within ±0.8 nm (red curve) with a sinusoidal shape, because higher order derivatives contribute some error.

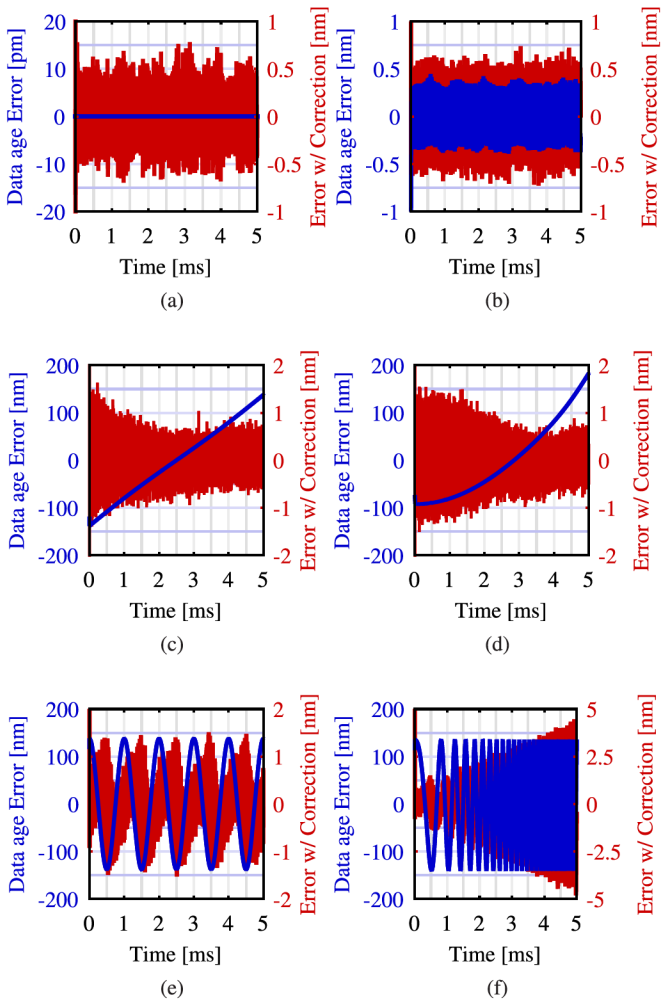


Fig. 17. Data age error (blue curve) and the residual error after compensation (red curve) for (a) static, (b) constant velocity, (c) linear increasing velocity, (d) linear increasing acceleration, (e) sinusoid velocity, and (f) sinusoid velocity with varied oscillating frequency scenarios.

Another simulation also assesses a sinusoid velocity motion, but the oscillating frequency is increasing from 0.5 to 5 kHz. Fig. 16(b) shows the errors before (blue curve) and after (red curve) compensation. When the oscillating frequency is high, the residual error after compensation becomes significant, which is produced by the higher order (≥3rd) derivatives, and cannot be ignored anymore

$$\sum_{n=3}^{\infty} \frac{\phi_d{}^{(n)}(t)}{n!} \tau^n \not\approx 0. \tag{28}$$

We examined this scenario as an extreme case. In fact, the oscillating frequency of our typical measurement scenario is much lower than 5 kHz. Compensating the first- and second-order error is sufficient for our measurements. For motion whose velocity changes at a rapid pace, more higher order derivatives must be kept to be accurate.

B. Variable Delay Time

If implementing the filter as IIR filters, a varied phase delay is introduced. The total equivalent time delay is varied as well. In this design, the fourth-order IIR filters are employed, whose cutoff frequency is 750 kHz. The overall time delay of the system is varied around 0.85 μs.

The same simulations are performed to verify the performance of compensating this varied time delay in Fig. 17. Similarly, the compensation module introduces some error itself, which is about ±0.6 nm. In the static [Fig. 17(a)] and constant velocity [Fig. 17(b)] scenarios, the error after compensation is larger than the original error.

In linear increasing velocity [Fig. 17(c)], linear increasing acceleration [Fig. 17(d)], and sinusoid velocity [Fig. 17(e)] scenarios, the data age error is within ±150 nm (blue curve), which is about one-fourth of that of FIR. Because the time delay introduced by the IIR is less than FIR, the data age error is less as well. However, the performance of the error compensation is not as well as that of FIR. The residual error is about ±1.5 nm (red curve), which is about two times of FIR. The reason is that the filtering performance of the fourth-order IIR filter is not as well as the 37-tap FIR filter, so the signal at frequency $2f_s + f_D$ is not completely suppressed.

In the sinusoidal velocity with varied oscillating frequency [Fig. 17(f)] scenario, the residual error becomes significant, because the compensation module still just compensates the first- and second-order error. With the oscillating frequency increasing, the higher order error cannot be ignored.

VII. Conclusion

The data age error becomes significant but often overlooked when there are rapid position changes and nonconstant velocity motions of the target, whose value could be up to hundreds of nanometers depending on the motion of the target and the time delay of the system.

In this paper, we have investigated the source of data age error, presented an upgraded model of this error, proposed methods for modeling and compensating it for heterodyne interferometry applications, and implemented the error compensation module in an FPGA. The synthesizable error compensation module consists of a derivative tracking part, and an error computation part.

The FPGA HIL simulation results show that this implementation can decrease the data age error from ±600 to ±0.8 nm for fixed time delay, and from ±150 to ±1.5 nm for the variable time delay in dynamic cases and still keep the sub-nanometer resolution in quasi-static cases. This increases the dynamic accuracy of displacement measurement, especially for nonconstant velocity motions.

References

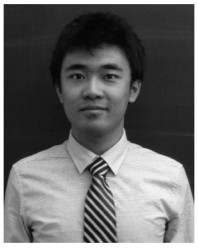
[1] F. C. Demarest, "High-resolution, high-speed, low data age uncertainty, heterodyne displacement measuring interferometer electronics," *Meas. Sci. Technol.*, vol. 9, no. 7, p. 1024, 1998.

[2] *Methods for Performance Evaluation of Computer Numerically Controlled (CNC) Lathes and Turning Centers*, ASME Standard B5.57-1998, 1998.

[3] *Methods for Performance Evaluation of Computer Numerically Controlled Machining Centers*, ASME Standard B5.54-2005, 2005.

[4] *Test Code for Machine Tools—Part 1: Geometric Accuracy of Machines Operating Under No-Load or Finishing Conditions*, ISO Standard 230-1:2012, 2012.

[5] R. R. Fesperman, Jr., *et al.*, "Methods for performance evaluation of single axis positioning systems: A new standard," in *Proc. 28th ASPE Annu. Meeting*, vol. 56. Saint Paul, MN, USA, Oct. 2013, pp. 498–503.
[6] C. Wang and J. D. Ellis, "Phase compensation for dynamic doppler frequency shifts," in *Proc. 29th ASPE Annu. Meeting*, Boston, MA, USA, 2014, pp. 1–2.
[7] J. D. Ellis, *Field Guide to Displacement Measuring Interferometry*. Bellingham, WA, USA: SPIE Press, 2013.
[8] C. Wang, "FPGA-based, 4-channel, high-speed phasemeter for heterodyne interferometry," M.S. thesis, Dept. Elect. Comput. Eng., Univ. Rochester, Rochester, NY, USA, 2013.
[9] C. Wang and J. D. Ellis, "Dynamic Doppler frequency shift errors: Measurement, characterization, and compensation," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1994–2004, Jul. 2015.
[10] V. G. Badami and P. J. de Groot, "Displacement measuring interferometry," in *Handbook of Optical Dimensional Metrology*, K. Harding, Ed. New York, NY, USA: Taylor & Francis, 2013.
[11] L. D. Cosart, L. Peregrino, and A. Tambe, "Time domain analysis and its practical application to the measurement of phase noise and jitter," *IEEE Trans. Instrum. Meas.*, vol. 46, no. 4, pp. 1016–1019, Aug. 1997.
[12] J.-J. Vandenbussche, P. Lee, and J. Peuteman, "On the accuracy of digital phase sensitive detectors implemented in FPGA technology," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 8, pp. 1926–1936, Aug. 2014.
[13] B. Djokic and E. So, "Phase measurement of distorted periodic signals based on nonsynchronous digital filtering," *IEEE Trans. Instrum. Meas.*, vol. 50, no. 4, pp. 864–867, Aug. 2001.
[14] X. Zhu, Y. Li, S. Yong, and Z. Zhuang, "A novel definition and measurement method of group delay and its application," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 1, pp. 229–233, Jan. 2009.
[15] M. R. Marcin, "Digital receiver phase meter for LISA," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2446–2453, Dec. 2005.
[16] P. Köchert, J. Flügge, C. Weichert, R. Köning, and E. Manske, "Phase measurement of various commercial heterodyne He–Ne-laser interferometers with stability in the picometer regime," *Meas. Sci. Technol.*, vol. 23, no. 7, p. 074005, 2012.
[17] F. C. Demarest, "Method and apparatus for providing data age compensation in an interferometer," U.S. Patent 5 767 972, Jun. 16, 1998.
[18] F. C. Demarest, "Data age adjustments," U.S. Patent 6 597 459, Jul. 22, 2003.
[19] L. C. Kalem and D. T. Dieken, "Laser measurement system with digital delay compensation," U.S. Patent 6 807 497, Oct. 19, 2004.
[20] F. C. Demarest, "Data age compensation with avalanche photodiode," U.S. Patent 7 542 147, Jun. 2, 2009.
[21] P. O'Shea, *The Measurement, Instrumentation, and Sensors: Handbook* (The Electrical Engineering Handbook Series). Boca Raton, FL, USA: CRC Press, 1999, pp. 41-1–41-19.
[22] H. Müller, S.-W. Chiow, Q. Long, C. Vo, and S. Chu, "Active sub-Rayleigh alignment of parallel or antiparallel laser beams," *Opt. Lett.*, vol. 30, no. 24, pp. 3323–3325, 2005.
[23] T. Schuldt, M. Gohlke, D. Weise, U. Johann, A. Peters, and C. Braxmaier, "Picometer and nanoradian optical heterodyne interferometry for translation and tilt metrology of the LISA gravitational reference sensor," *Classical Quantum Gravity*, vol. 26, no. 8, p. 085008, 2009.
[24] S. R. Gillmer, R. C. G. Smith, S. C. Woody, and J. D. Ellis, "Compact fiber-coupled three degree-of-freedom displacement interferometry for nanopositioning stage calibration," *Meas. Sci. Technol.*, vol. 25, no. 7, p. 075205, 2014.
[25] X. Yu, S. R. Gillmer, and J. D. Ellis, "Beam geometry, alignment, and wavefront aberration effects on interferometric differential wavefront sensing," *Meas. Sci. Technol.*, vol. 26, no. 12, p. 125203, 2015.
[26] S. L. Mielke and F. C. Demarest, "Displacement measurement interferometer error correction techniques," in *Proc. ASPE Topical Meeting Precis. Mech. Design Mechatronics Sub-50nm Semiconductor Equip.*, vol. 43. Berkeley, CA, USA, 2008, pp. 113–116.
[27] H. Sira-Ramírez, C. G. Rodríguez, J. C. Romero, and A. Juárez, *Algebraic Identification and Estimation Methods in Feedback Control Systems*. New York, NY, USA: Wiley, Mar. 2014.
[28] R. Chartrand, "Numerical differentiation of noisy, nonsmooth data," *ISRN Appl. Math.*, vol. 2011, Apr. 2011, Art. no. 164564.
[29] I. Knowles and R. J. Renka, "Methods for numerical differentiation of noisy data," *Electron. J. Differ. Equ.*, vol. 21, pp. 235–246, Feb. 2014.
[30] J. Reger, H. S. Ramirez, and M. Fliess, "On non-asymptotic observation of nonlinear systems," in *Proc. 44th IEEE Conf. Decision Control*, Dec. 2005, pp. 4219–4224.
[31] R. Morales, F. Rincón, J. D. Gazzano, and J. C. López, "Real-time algebraic derivative estimations using a novel low-cost architecture based on reconfigurable logic," *Sensors*, vol. 14, no. 5, pp. 9349–9368, 2014.
[32] M. Tanaka, T. Yamagami, and K. Nakayama, "Linear interpolation of periodic error in a heterodyne laser interferometer at subnanometer levels [dimension measurement]," *IEEE Trans. Instrum. Meas.*, vol. 38, no. 2, pp. 552–554, Apr. 1989.
[33] C. Wang and J. D. Ellis, "Real-time periodic error correction for heterodyne interferometers using Kalman filters," in *Proc. 30th ASPE Annu. Meeting*, Austin, TX, USA, 2015, pp. 409–414.
[34] "Hardware in the loop from the MATLAB/simulink environment," Altera Corp., San Jose, CA, USA, White Paper WP-01208-1.0, Sep. 2013.



Chen Wang (GSM'13) received the B.E. degree in measurement, control technique, and instruments from the Harbin Institute of Technology, Harbin, China, in 2011, and the M.S. degree in electrical engineering from the University of Rochester, Rochester, NY, USA, in 2013. He currently is a Ph.D. candidate in electrical engineering at University of Rochester.

His current research interests include the high-speed, real-time, precise electrical/electronic measurement and control system design, mainly including signal processing, dynamic control algorithm analysis, and development and implementation of field-programmable gate array and embedded system.

Jonathan D. Ellis (M'14) received the B.S. and M.S. degrees from the University of North Carolina at Charlotte, Charlotte, NC, USA, and the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, all in mechanical engineering.

He is currently an Assistant Professor with the University of Rochester, Rochester, NY, USA, with a joint appointment between the Department of Mechanical Engineering and The Institute of Optics. His current research interests include dimensional metrology, precision instrumentation, and instrumentation for biological application.