

RESEARCH ARTICLE | DECEMBER 26 2018

## Real time implementation of empirical mode decomposition algorithm for ultrasonic nondestructive testing applications



Eaglekumar G. Tarpara ; V. H. Patankar



*Rev. Sci. Instrum.* 89, 125118 (2018)

<https://doi.org/10.1063/1.5074152>



View  
Online



Export  
Citation

CrossMark

**Customize your own Lock-in Amplifier**



[www.ssi-instrument.com](http://www.ssi-instrument.com)

PXIe module Lock-in Amplifier  
Multi-channel Lock-in Amplifier  
Up to 8 demodulators  
Toolset: Scope, FFT, PID, Sweeper



DC to 300MHz frequency

# Real time implementation of empirical mode decomposition algorithm for ultrasonic nondestructive testing applications

Eaglekumar G. Tarpara<sup>1,a)</sup> and V. H. Patankar<sup>1,2</sup>

<sup>1</sup>Homi Bhabha National Institute (HBNI), Mumbai 400094, India

<sup>2</sup>Bhabha Atomic Research Centre (BARC), Mumbai 400085, India

(Received 22 October 2018; accepted 5 December 2018; published online 26 December 2018)

A real time empirical mode decomposition (EMD) algorithm based ultrasonic imaging system has been developed for non-destructive testing (NDT) applications. It is difficult to implement the EMD based signal processing algorithm in real time because it is totally a data-driven process which comprises numerous sifting operations. In this paper, the EMD algorithm has been implemented in the visual software environment. The EMD implementation encompasses two types of interpolation methods: piecewise linear interpolation (PLI) and cubic spline interpolation (CSI). The cubic spline tridiagonal matrix has been solved by using the Thomas algorithm for real time processing. The total time complexity functions for both the implemented PLI and CSI based EMD methods have been computed. For the signal filtering, the partial reconstruction algorithm has been adopted. The baseline correction and noise filtering applications have been presented using an EMD based visual software. The real time practicability and the efficiency of this method have been validated through ultrasonic NDT experimentation for improvement in the time domain resolution of the ultrasonic A-scan raw data. The practical results show that in the noisy environment, it is possible to enhance the signal-to-noise ratio for the visualization and identification of ultrasonic pulse-echo signals in real time. *Published by AIP Publishing.* <https://doi.org/10.1063/1.5074152>

## I. INTRODUCTION

Ultrasonic inspection is one of the most successful and widely used non-destructive testing (NDT) technique for quality assessment, identification, detection, and characterization of flaws in metal structures. The signal reflected by discontinuities/flaws includes the information about the anomalies, based on which we can identify its location, size, and category. In order to detect small planer cracks or flaws in highly scattered or coarse grain materials such as stainless steel, the noise from the grain boundaries called grain noise and backscattering noise play a major role as this noise value can be higher than the desired flaw echoes. Thus, the detection of flaws in the presence of noise becomes very difficult.<sup>1</sup> The same type of effect is observed while carrying out the time-of-flight diffraction (TOFD) technique. However, electronics front-end hardware of the ultrasonic instrumentation also introduces the noise in the feeble echo signals. Most of the noise is introduced due to the AC power-line interference, improper impedance matching, and high-frequency interference.<sup>2</sup> In experimental science, noise equals any random fluctuation in data which obstructs discernment of the desired signal. Thus, the main aim is the noise suppression and echo identification of ultrasonic pulse-echo signals in real time using the developed experimental setup under the noisy environment.

As the ultrasonic echo is non-linear and non-stationary, to process/denoise this kind of signal, various signal processing techniques have been used. But every technique has some advantages and drawbacks. For example, cross correlation is

the easy method for the implementation, but if the noise and signal have the same amplitude in the same frequency range, then it would be impossible to discriminate between them.<sup>3</sup> The linear filtering methods such as Wiener filtering<sup>4</sup> are also easy to design and implement, but they are not effective when the signal contains sharp edges and impulses of short period. The non-linear methods, such as wavelet transform,<sup>5,6</sup> are most popular due to their time and frequency localization properties. But it requires the selection of the mother wavelet function and the scale of it for signal enhancement.

In this paper, the empirical mode decomposition (EMD), proposed by Huang *et al.*,<sup>7</sup> is utilized for the signal analysis and processing in the time domain. This data-driven decomposition algorithm is used for de-noising of non-linear and non-stationary ultrasonic data sample sequences. EMD decomposes the signal into a series of narrow-band oscillatory components with different features and a residue function. Each oscillatory component is called an intrinsic mode function (IMF). Different from other methods such as wavelet, the EMD does not require a pre-defined basis function or mother wavelet for the signal decomposition. The process of decomposition using EMD is totally non-model based data-driven.

The EMD requires many iterative calculations, and thus it cannot be implemented using parallel architecture.<sup>8</sup> Many real-life applications require real time results such as detection of faults in operating machines, real time inspection of material integrity using NDT, etc. However, real time fast implementation of EMD is a major challenge. In the past few years, many authors have presented EMD based real time implementation in hardware, software, and hardware-software mix environment.<sup>8–13</sup> Ultrasonic noise suppression and signal

<sup>a)</sup>Electronic mail: eagle252525@gmail.com

identification using non-real time EMD have been described in Refs. 14 and 15. The EMD based algorithm should be performed in real-time, especially for applications (or instruments) where online visual inspections of signals/waveforms are mandatory such as NDT instruments, ECG instruments, digital oscilloscope, etc. In this article, an online implementation of an EMD based filtering scheme has been presented for better visualization and identification of ultrasonic pulse-echo signals in real-time.

## II. EMD AND SIGNAL RECONSTRUCTION

### A. EMD algorithm

The original signal  $x(t)$  is decomposed into the small number of IMFs and a residue function. The IMFs and finite residue sequences are produced through the shifting process of the signal. Using the EMD, the original signal  $x(t)$  can be decomposed as<sup>7</sup>

$$x(t) = \sum_{i=1}^{n_f} c_i(t) + r(t), \quad (1)$$

where  $n_f$  denotes the number of extracted IMFs and  $c_i(t)$  and  $r(t)$  are the IMF and the finite residue of the original signal  $x(t)$ , respectively.

The steps of the EMD procedures are described as follows:

1. Let  $i$  and  $j$  denote the inner and outer iterative loop indices, respectively. Initially, assume  $x_{i,j}(t) = x(t)$ .
2. Identify all the local maxima and minima of the signal  $x_{i,j}(t)$ .
3. Interpolate these local minima and maxima points by CSI or PLI and generate lower envelope  $le(t)$  and upper envelope  $ue(t)$ .
4. Calculate the mean envelope  $mean(t)$  by averaging both envelopes,  $mean_{i,j}(t) = [le(t) + ue(t)]/2$ .
5. Take the difference between data  $x_{i,j}(t)$  and  $mean_{i,j}(t)$ ,  $h_{i,j}(t) = x_{i,j}(t) - mean_{i,j}(t)$ .
6. Check if  $h_{i,j}(t)$  accomplishes conditions defining an IMF. If  $h_{i,j}(t)$  is not an IMF, then it is treated as original data  $x_{i+1,j}(t) = h_{i,j}(t)$  and repeat the step 1 to 5. If  $h_{i,j}(t)$  is an IMF, then save the IMF component  $c_j(t) = h_{i,j}(t)$ .
7. Obtain the remaining residue function  $r_j(t)$  by using  $r_j(t) = x_{i,j}(t) - c_j(t)$ .
8. If residue  $r_j(t)$  satisfies the EMD termination criterion, i.e., the number of extrema in  $r_j(t) < 2$ , then stop the shifting process and  $r_j(t)$  becomes the final residue function  $r(t)$ ; otherwise, treat  $r_j(t)$  as a new set of data  $x_{i,j+1}(t) = r_j(t)$ , and repeat the step 1 to 7 to find other IMFs  $c_1(t), c_2(t), \dots, c_{n_f}(t)$ .

Thus, the original signal  $x(t)$  is decomposed into the  $n_f$  IMFs  $c_1(t), c_2(t), \dots, c_{n_f}(t)$  and a final residue function  $r(t)$ .

### B. Signal reconstruction using EMD

The filtering scheme depends on the basic idea that most of the signals are usually concentrated on lower frequency components (last IMFs) and decrease toward high-frequency modes (first IMFs). For the signal that is corrupted by white

Gaussian noise, signal-to-noise ratio (SNR) is higher at low frequencies than at the higher ones. Here we have adopted the partial filtering algorithm proposed by Boudraa and Cexus<sup>16</sup> because it is a fully data-driven approach, and hence it does not require any pre-possessing and post-processing of data. Thus, the actual signal  $x_a(t)$  superimposed by white Gaussian noise  $n_g(t)$  is described as

$$x(t) = x_a(t) + n_g(t). \quad (2)$$

The main objective is to find an approximation  $\tilde{x}_a(t)$  from the noisy signal  $x(t)$ . There is a certain mode index  $k$ , after which the  $c_k$  ( $k$ th IMF) allows us to retrieve the most of the information of the actual signal from the noisy signal. Thus, the modes after  $c_k$  dominate the signal, whereas the previous modes contain high-frequency components. So for the signal reconstruction, high-frequency dominated components will not be used. The signal  $\tilde{x}_a(t)$  is reconstructed using selected  $(n_f - k + 1)$  IMFs, and, from (1), it is given as

$$\tilde{x}_a^k(t) = \sum_{i=k}^{n_f} c_i(t) + r(t), \quad (3)$$

where  $k = 2, 3, \dots, n_f$ . The consecutive mean square error (CMSE) has been used, which does not require any information of  $x_a(t)$ . The CMSE measures the squared Euclidean length between two consecutive reconstructions of the signal<sup>16</sup>

$$CMSE(\tilde{x}_a^k, \tilde{x}_a^{k+1}) = \frac{1}{N} \sum_{i=1}^N [\tilde{x}_a^k(i) - \tilde{x}_a^{k+1}(i)]^2 = \frac{1}{N} \sum_{i=1}^N [c_k(i)]^2, \quad (4)$$

where  $k = 1, 2, \dots, n_f - 1$  and  $N$  is the data length of the signal. Here, the CMSE is reduced to the energy of the  $k$ th IMF. The main aim of the signal reconstruction is to find the optimal index  $k = k_c$  after which CMSE has the minimum value or it has significant change in the IMF energy. It is given as

$$k_c = \underset{1 \leq k \leq n_f - 1}{\operatorname{argmin}} [CMSE(\tilde{x}_a^k, \tilde{x}_a^{k+1})]. \quad (5)$$

Thus, the steps for the signal reconstruction are as follows: (1) decompose the signal  $x(t)$  using (1), (2) compute  $CMSE(\tilde{x}_a^k, \tilde{x}_a^{k+1})$ , for  $k = 1, 2, \dots, n_f - 1$  using (4), (3) find optimal value  $k = k_c$  using (5), and (4) reconstruct the filtered signal  $\tilde{x}_a(t)$  using (3).

## III. ONLINE IMPLEMENTATION OF EMD BASED SIGNAL PROCESSING ALGORITHM

### A. Envelope generation

The local extrema extraction is the first step of the EMD algorithm. We adopted the first derivative method to extract the local minima and maxima points. The function  $F(x)$  has

Local maxima at  $c$ , if  $F'(c - 1) > 0$  and  $F'(c) < 0$

Local minima at  $c$ , if  $F'(c - 1) < 0$  and  $F'(c) > 0$

No local extrema at  $c$ , if otherwise.

An appropriate interpolation algorithm must be implemented to connect the local extrema points to form an envelope. In this design, two interpolation methods, PLI and CSI, are considered for envelope generation. Figure 1 shows an

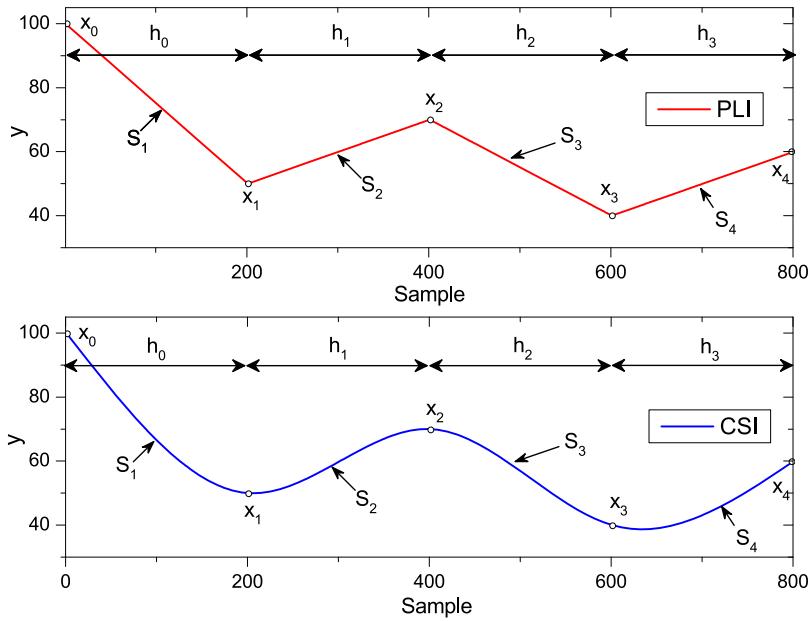


FIG. 1. PLI and CSI curves connecting to 5 data points,  $x_0$  to  $x_4$ .

example of 5 points and its generated PLI and CSI curves connecting them all together.

The piecewise polynomial interpolation (PPI) splits the data points  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$  into a set of intervals for  $j = 0, 1, \dots, n - d$  and constructs a separate polynomial of degree  $d$  for all intervals. The full curve  $S(x)$  is expressed as  $S(x) = \sum_{i=1}^n S_i(x)$ . Each  $S_i(x)$  is the polynomial in the interval  $[x_{i-1}, x_i]$ . The PLI (degree  $d = 1$ ) connects pairs of data points with a straight linear line  $S_i(x)$ , and it is given as

$$S_i(x) = \begin{cases} y_k + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}(x - x_k), & \text{if } x_k \leq x \leq x_{k+1} \\ 0, & \text{if otherwise} \end{cases}. \quad (6)$$

This method of interpolation can be extended by constructing a new set of interpolating polynomial  $S_i(x)$  of higher degree ( $d = 3$ ). To derive mathematical formulas of the cubic spline, assumed extrema points are  $a = x_0 < x_1 < \dots < x_n = b$  and the difference is  $h_i = |x_{i+1} - x_i|$ . The cubic spline is a third degree cubic polynomial, and it is given as

$$\text{插值函数 } S_{i+1}(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad (7)$$

where  $i = 0, 1, \dots, n - 1$ . The interpolation data points impose  $n + 1$  constraints. The cubic spline 8 properties supply additional  $n + 1$  constraints.<sup>10</sup> Thus,  $n + 1 + 3(n - 1) = 4n - 2$  constraints are specified for the  $4n$  unknown coefficients ( $a_i$ ,  $b_i$ ,  $c_i$ , and  $d_i$ ). For the extra two boundary conditions, end slope spline conditions  $S'_0(x_0) = y_0'$  and  $S'_{n-1}(x_n) = y_n'$  are used. It is also called the complete cubic spline condition. The first derivative values of the data may not be readily available, but they can be replaced by exact approximations. By using these conditions, following relations hold for  $i = 0, 1, \dots, n - 1$ :

$$\begin{aligned} a_i &= \frac{m_{i+1} - m_i}{6h_i}, \\ b_i &= \frac{m_i}{2}, \\ c_i &= \frac{y_{i+1} - y_i}{h_i} - \frac{2m_i h_i + h_i m_{i+1}}{6}, \\ d_i &= y_i. \end{aligned} \quad (8)$$

Here,  $h_i$  and  $y_i$  are the known values from the data points, so all that remains is to solve for  $m_i$ . After combining all the equations, we can obtain the system of linear equation  $A = mB$ , where

$$A = \begin{bmatrix} a_{00} & a_{01} & 0 & \cdots & 0 & 0 & 0 \\ a_{10} & a_{11} & a_{12} & \cdots & 0 & 0 & 0 \\ 0 & a_{21} & a_{22} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix},$$

$$\begin{aligned} a_{00} &= 2h_0 & a_{22} &= 2(h_1 + h_2) \\ a_{01} &= h_0 & a_{n-1,n-2} &= h_{n-2} \\ a_{10} &= h_0 & a_{n-1,n-1} &= 2(h_{n-2} + h_{n-1}) \\ a_{11} &= 2(h_0 + h_1) & a_{n-1,n} &= h_{n-1} \\ a_{12} &= h_1 & a_{n,n-1} &= h_{n-1} \\ a_{21} &= h_1 & a_{nn} &= 2h_{n-1} \end{aligned}, \quad (9)$$

$$m = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{bmatrix}, B = \begin{bmatrix} \frac{6}{h_0}(y_1 - y_0) - 6y'_0 \\ \frac{6}{h_1}(y_2 - y_1) - \frac{6}{h_0}(y_1 - y_0) \\ \frac{6}{h_2}(y_3 - y_2) - \frac{6}{h_1}(y_2 - y_1) \\ \vdots \\ \frac{6}{h_{n-1}}(y_n - y_{n-1}) - \frac{6}{h_{n-2}}(y_{n-1} - y_{n-2}) \\ 6y'_n - \frac{6}{h_{n-1}}(y_n - y_{n-1}) \end{bmatrix}.$$

Here,  $A$  is the tridiagonal matrix. In order to solve the tridiagonal system of equations  $A = mB$ , we adopted the **TriDiagonal Matrix Algorithm** (TDMA), also known as the Thomas algorithm, which is a simplified form of Gaussian elimination.<sup>17</sup> The Thomas algorithm proceeds in two stages. At the first stage, the tridiagonal system is transformed into a bidiagonal system called forward elimination. At the second stage, this bidiagonal system is solved by back substitution. So the

total computations required by using this algorithm is only  $\sim O(n)$ , in contrast to the computation cost  $O(n^3)$  required for the general Gaussian elimination algorithm. The pseudo code for the TDMA (Thomas algorithm) is as follows:

```

Step 1. Formation of  $[A/B] = m$  system of linear equations
Step 2. For  $i = 1$  to  $n$  Do
     $t = A[i + 1, i]/A[i, i]$ 
    For  $j = 1$  to  $i + 1$  Do
         $A[i + 1, j] = A[i + 1, j] - t * A[i, j]$ 
    End For
End For
Step 3.  $m[n] = A[n + 1, n + 1]/A[n + 1, n + 2]$ 
For  $i = n - 1$  to 0 Do
    sum = 0
    For  $j = 1$  to  $i - 1$  Do
        sum = sum +  $A[i, j] * m[j]$ 
    End For
     $m[i] = (A[i, n] - sum)/A[i, i]$ 
End For

```

## B. IMF stoppage shifting criteria

To get physically meaningful IMFs, the stoppage shifting criteria should be defined accordingly. Previously, various stoppage criteria have been proposed such as the Cauchy type criterion,<sup>7</sup> the mean value criterion,<sup>18</sup> the S-number criterion,<sup>19</sup> and the fixed shifting time criterion.<sup>20</sup> We have chosen the Cauchy type shifting criteria because it is easy to implement. The IMF shifting process will stop when the SD (standard deviation) is smaller than the fixed predefined value  $\zeta$

$$SD = \sum_{i=1}^N \frac{|d_{j(k-1)}(i) - d_{jk}(i)|}{d_{j(k-1)}^2(i)} \leq \zeta, \quad (10)$$

where  $N$  is the total data length,  $\zeta$  is the threshold number ( $\zeta = 0.3$ ), and  $d_{jk}(i)$  is the  $k$ th shifting result of the inner shifting loop of  $j$ th mode.

## IV. TIME COMPLEXITY OF IMPLEMENTED EMD

In this section, we have analyzed the time complexity of the implemented EMD algorithm. The previously published work<sup>21</sup> does not encompass the computations of IMF stoppage criteria and signal reconstruction. The arithmetic operation includes addition/subtraction (ADD), multiplication (MUL), division (DIV), and comparison (COMP). Here, we

have used the float data type for the arithmetic data operations. The computational time complexity of the both PLI and CSI based EMD has been calculated. The main procedures like extrema identification, IMF calculations, mean envelope calculations, checking of IMF stopping criteria, etc., are identical for both types of EMD. The detailed descriptions are provided below.

Let each of the input  $x(t)$ , pre-IMF, post-IMF  $c(t)$ , residue  $r(t)$ , mean, and upper/lower envelope  $ue(t)/le(t)$  has equal length  $n$ . The decomposition results of EMD consist of total  $n_f$  IMFs each with length  $n$ . So it requires total  $n \cdot n_f$  storage for all IMFs. The inner loop of EMD comprises multiple iterations for every unsatisfied IMF stoppage criteria. Thus, the inner loop requires total  $n_{pf} \cdot n_f$  iterations, where  $n_{pf}$  is the number of each inner loop iteration. Assume, there are  $n_{ue}(f)$  maxima and  $n_{le}(f)$  minima points for each IMF. So the total number of extrema for each IMF is  $n_e(f) = n_{ue}(f) + n_{le}(f)$ . The optimal mode number of an IMF is represented by  $n_k$  using (5). The total time complexity of implemented EMD functions is provided in Table I. Here, it is assumed that all arithmetic operations require the same amount of time for the numerical calculations. The total amount of time count for the PLI based EMD from Table I is

$$T_{PLI-EMD} = T_{shift} + T_{stop} + T_{extr} + T_{PLI} + T_{reco}, \\ T_{PLI-EMD} = (11n_{pf} + 4)n_f n + 1n_k n + 9 \sum_{f=1}^{n_f} n_e(f)n_{pf}. \quad (11)$$

Here,  $\max_{1 \leq k \leq n-1} (n_k) = n_f$ . So (11) can be written as

$$T_{PLI-EMD} = \left[ 11n_f n \left( 1 + \frac{5}{11n_{pf}} \right) \right] n_{pf} + 9 \sum_{f=1}^{n_f} n_e(f)n_{pf}. \quad (12)$$

Since  $11n_{pf} \gg 5$ , (12) can be reduced as

$$T_{PLI-EMD} = \left[ 11n_f \cdot n + 9 \sum_{f=1}^{n_f} n_e(f) \right] \cdot n_{pf} \quad (13)$$

As mentioned earlier, CSI envelope generation is divided into three stages: formation of a tridiagonal matrix, forward elimination, and backward substitution. All arithmetic operations repeat for each IMF, and hence the total time complexity function for the matrix formulation is  $T_{math} = (5ADD + 2MUL + 1DIV)n_{pf} \sum n_e(f)$ . The time complexity function for the forward elimination is  $T_{for} = (2ADD + 2MUL + 1DIV)n_{pf} \sum n_e(f)$ . The same way for the backward substitution, the time complexity function is  $T_{back} = (2ADD + 1MUL + 1DIV)n_{pf} \sum n_e(f)$ . From (8), the numbers of required operations for the coefficient

TABLE I. Time complexity of EMD functions.

Function	$T$	Time complexity
Shifting	$T_{shift}$	$(3ADD + 1DIV)n_{pf} \cdot n_f \cdot n$
IMF stoppage criteria	$T_{stop}$	$(2ADD + 1MUL + 1DIV + 1COMP)n_{pf} \cdot n_f \cdot n$
Extrema identification	$T_{extr}$	$[(1ADD + 1COMP)n_f \cdot n + (1COMP) \sum_{f=1}^{n_f} n_e(f)] \cdot n_{pf}$
PLI envelope generation	$T_{PLI}$	$(4ADD + 2DIV + 2COMP) \cdot n_{pf} \sum_{f=1}^{n_f} n_e(f)$
CSI envelope generation	$T_{CSI}$	$[(6ADD + 6MUL)n_f \cdot n + (13ADD + 9MUL + 7DIV) \sum_{f=1}^{n_f} n_e(f)] \cdot n_{pf}$
Signal reconstruction	$T_{reco}$	$(1ADD + 1MUL + 1DIV + 1COMP)n_f \cdot n + (1ADD)n_k \cdot n$

calculations are  $T_{coef} = (4ADD + 4MUL + 4DIV)n_{pf}\sum n_e(f)$ . The total number of required arithmetic calculations for the spline curve generation is  $T_{cubic} = (6ADD + 4MUL)n_{pf} \cdot n_f \cdot n$ . Like (12), the total time complexity of the implemented CSI based EMD algorithm can be calculated as

$$T_{CSI-EMD} = T_{CSI} (= T_{mat} + T_{for} + T_{back} + T_{coef} + T_{cubic}) + T_{shift} + T_{stop} + T_{extr} + T_{reco}, \quad (14)$$

$$T_{CSI-EMD} = \left[ 23n_f \cdot n \left( 1 + \frac{5}{23n_{pf}} \right) \right] n_{pf} + 30 \sum_{f=1}^{n_f} n_e(f) \cdot n_{pf}. \quad (15)$$

Since  $23n_{pf} \gg 5$ , (15) can be reduced as

$$T_{CSI-EMD} = \left[ 23n_f \cdot n + 30 \sum_{f=1}^{n_f} n_e(f) \right] \cdot n_{pf}. \quad (16)$$

Since  $\max_{1 \leq f \leq n_f} n_e(f) \approx n$ , (13) and (16) can be written as

$$T_{PLI-EMD} \approx 20 \cdot n_{pf} \cdot n_f \cdot n, \\ T_{CSI-EMD} \approx 53 \cdot n_{pf} \cdot n_f \cdot n. \quad (17)$$

The EMD behaves as a dynamic filter bank in which the Fourier spectra of all IMFs fall into the single step along the axis of period logarithm.<sup>22</sup> So for the  $n$  number of data samples, the total number of pre-IMFs required for the EMD algorithm is  $n_f \leq \log_2 n$ .<sup>23</sup> Assume that the number of post-IMF  $n_{pf}$  is fixed. Thus, the reduced time complexity functions are given as

$$T_{PLI-EMD} \leq 20 \cdot n_{pf} \cdot (\log_2 n) \cdot n, \\ T_{CSI-EMD} \leq 53 \cdot n_{pf} \cdot (\log_2 n) \cdot n. \quad (18)$$

Thus, the total time complexity for the PLI and CSI based EMD is equal to the  $O(n \log n)$ .

## V. EXPERIMENTAL SETUP

The presented (Fig. 2) real time EMD based ultrasonic imaging system for NDT application consists of four main sections: (1) test setup for NDT, (2) analog front-end electronics boards, (3) digital section (boards), and (4) GUI software for the data processing and waveform displaying. The experimental setup is shown in Fig. 3.

### A. Analog front-end hardware development

As shown in Fig. 2, this section consists of a high voltage (500 V) ultrasonic spike pulser for generation of high voltage negative spike pulse, an ultrasonic receiver amplifier for amplification of echo signals, and analog to digital converter for digitization of echo signals. A fast CMOS RF gate driver is used to drive the power MOSFET. A high-speed dual MOSFET driver (EL7202, Intersil Corp., SA) is used to enhance the drive capability for MOSFET. The n-channel enhancement switch mode MOSFET (IXZ308N120, IXYS RF, CA) is used for high voltage switching operation. The pulser board is triggered by +3.3 V rising edge pulse which generates -500 V spike impulse with 100 ns duration. The pulse repetition frequency is chosen as 1 kHz. The receiver stage consists of four ultrafast rectifying diodes (MUR4100EG, On Semiconductor, Pheonix) based bridge limiter circuit which provides protection to further stage electronics from the high voltage spike signals

### SETUP

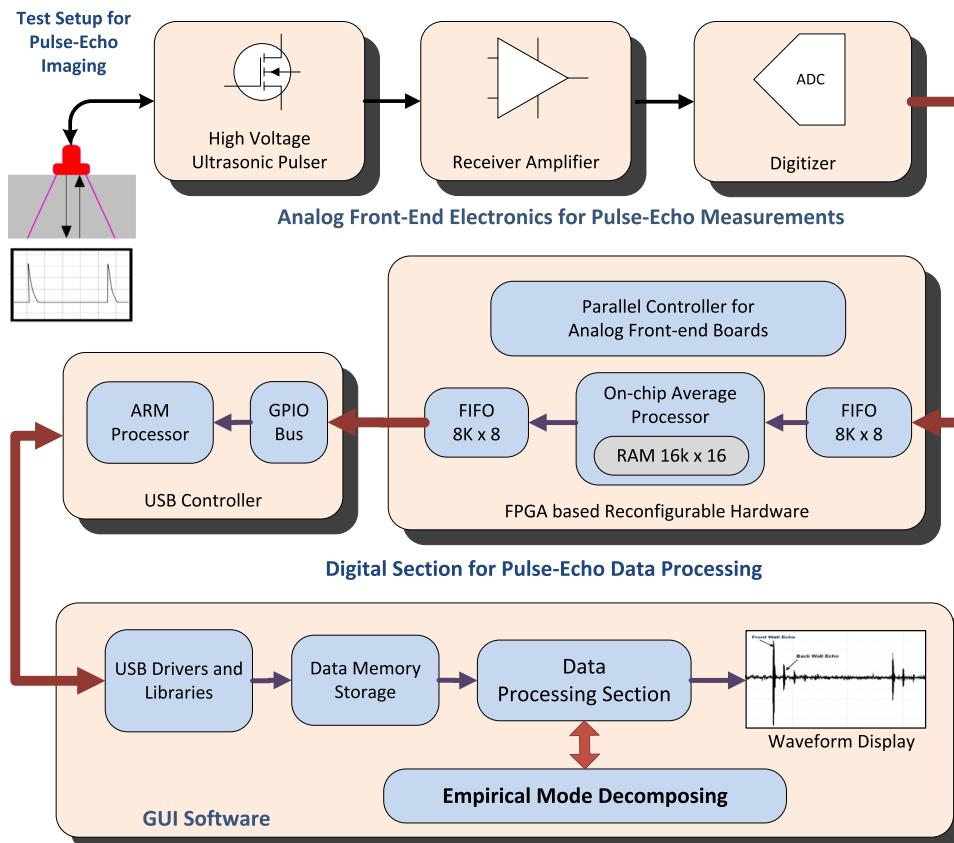


FIG. 2. Schematic block diagram of developed experimental setup for ultrasonic pulse-echo measurements.

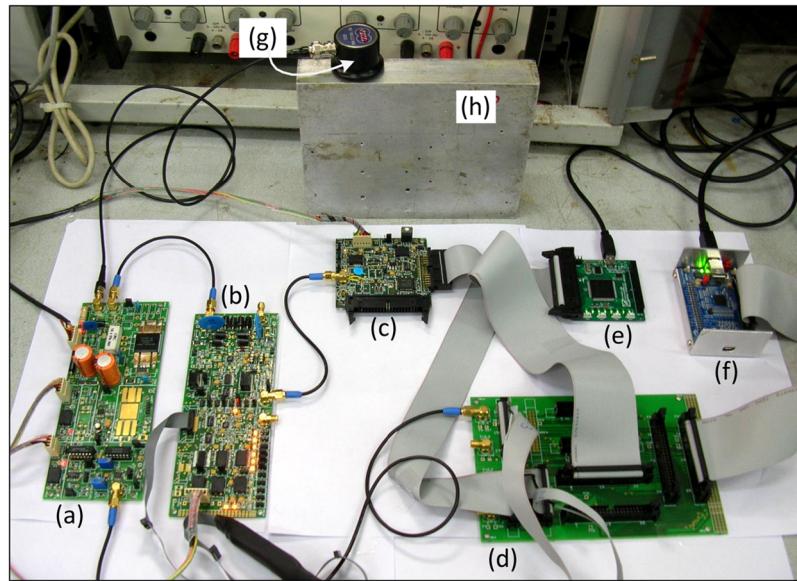


FIG. 3. Experimental setup: (a) high voltage ultrasonic pulser, (b) receiver amplifier, (c) digitizer board, (d) interface board, (e) FPGA module, (f) USB controller, (g) ultrasonic transducer, and (h) aluminum test setup.

received from the pulser circuit. The 400 MHz bandwidth, low power, high-performance amplifiers (AD8014, Analog Devices Inc.) are used as a non-inverting feedback amplifier with an individual fixed gain of +20 dB. The output of the cascaded amplifiers is connected to the analog multiplexer amplifier (AD8174, Analog Devices, Inc.) which offers a high speed disable feature allowing the output to be put into

a high impedance state. The 8-Bit, 100 MSPS, low power analog to digital converter (ADC) (AD9283, Analog Devices, Inc.) is utilized for the digitization.

## B. Digital section and GUI software for data processing

ADC is parallel interfaced with the Spartan-6 (XC6SLX9-3TQG114, Xilinx, Inc., San Jose, CA) FPGA. The 8-bit

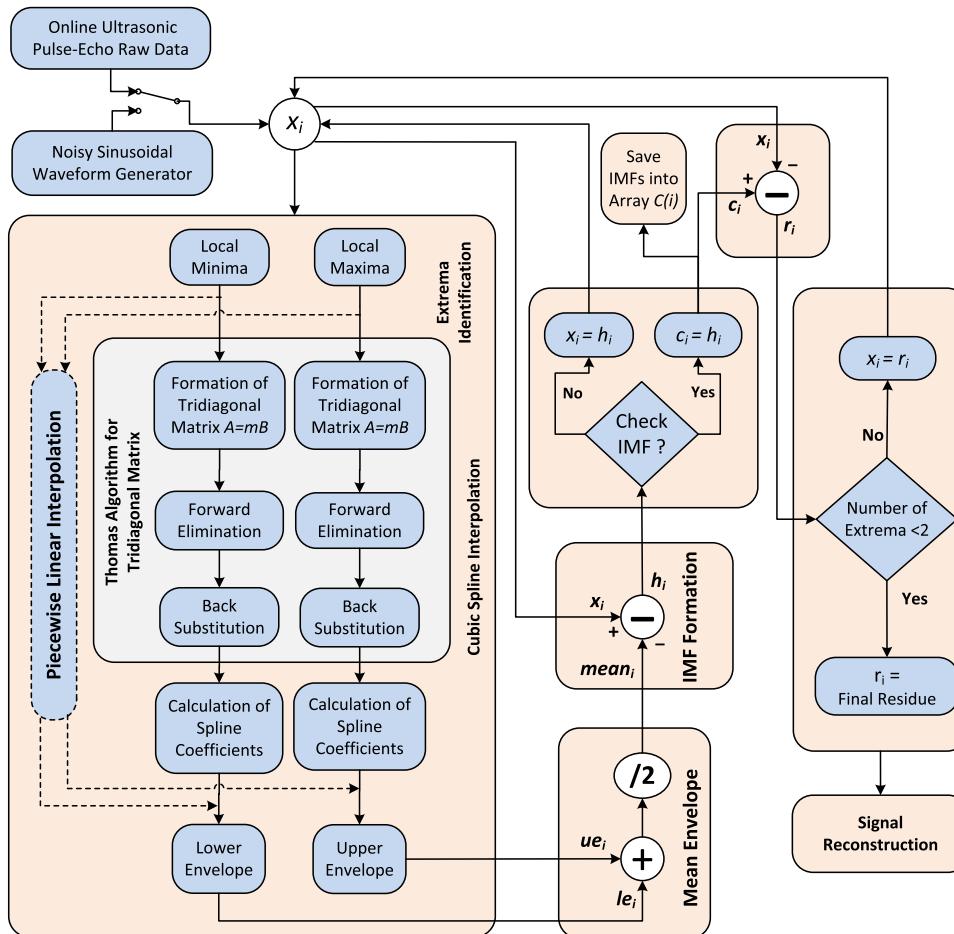


FIG. 4. Block diagram of implemented EMD based architecture.

output data of the ADC are stored in the on-chip first in, first out (FIFO) memory configured inside the FPGA for further processing of data. Data averaging hardware has been implemented in FPGA for the coherent averaging of each A-scan raw data packet. For coherent averaging, an on-chip dual port RAM of size 16k-16 bit has been developed for the data storage operations of intermediate stages. The averaged output data are again stored in the 8k-8 bit FIFO memory. The reconfigurable data controller architecture has been developed in FPGA by hand coding of VHDL code; these codes are compiled, synthesized, and fitted using Xilinx ISE Design Suite 14.7. A USB 3.0 peripheral controller (EZ-USB FX3, Cypress Semiconductor, San Jose, CA) with an on-chip 32-bit, 200-MHz ARM926EJ-S core central processing unit (CPU) has been used as a controller between FPGA and PC. The USB application firmware has been developed using the C++ language.

The new generation of application development tools has an even simpler, more powerful application programming interface (API). The SuiteUSB.net 2.0 supports the cyusb.sys device driver which increases the range of devices that can be accessed with the tool. The main element of the SuiteUSB.net is cyusb.dll. This DLL is a managed .NET 2.0 class library. The GUI software has been developed in Visual Studio platform using visual C# language. It encompasses the EMD based implementation as depicted in Fig. 4

## VI. RESULTS AND DISCUSSIONS

### A. Decomposition results by implemented EMD software

This section provides the decomposition capability of the implemented EMD algorithm using sifting operations to split the signal into its components with different frequencies. The numerical equation of the sinusoidal signal has been implemented in a visual software environment using C#, and it is described as  $x(i) = 70 \sin(2 \cdot \pi \cdot 0.01i) + 50 \sin(2 \cdot \pi \cdot 0.15i)$ ,

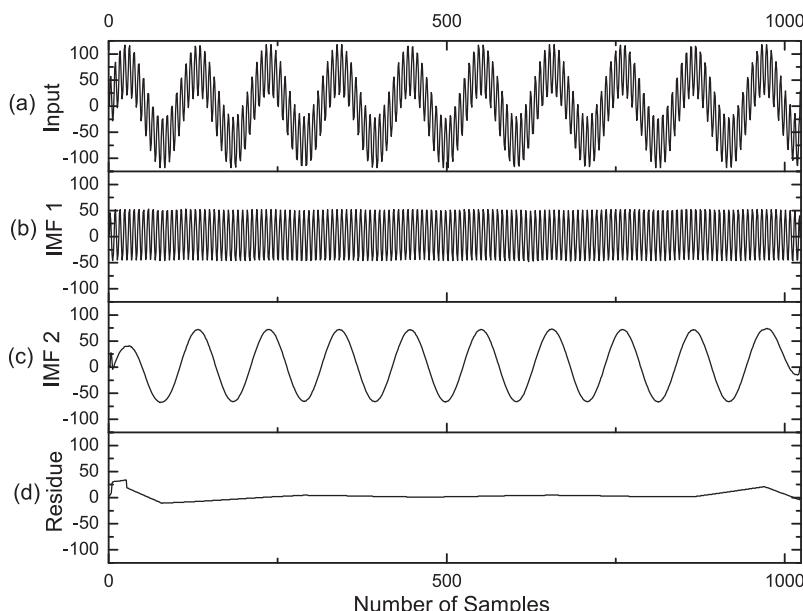


FIG. 5. Decomposition results by CSI-EMD: (a) Input, (b) IMF1, (c) IMF2, and (d) residue  $r$ .

$0 < i \leq 1024$ . Here,  $i$  is the number of samples, and the data acquisition/sampling rate is chosen as 10 MSPS. So the signal  $x(t)$  is composed of two different frequency components 1500 kHz and 100 kHz. The implemented sinusoidal input has been plotted, as shown in Fig. 5. It shows that the signal has been decomposed in IMF1 (1500 kHz), IMF2 (100 kHz), and residue function  $r$ . It can be observed that IMF2 and residue  $r$  have end-effect errors at both extreme ends because in most cases, the first and last endpoints of the signal are not the extreme points, and this causes upper/lower envelope become divergent at both end parts. This leads to the waveform distortion at the ends of the decomposed signal.

### B. Baseline correction and noise filtering by EMD

The signal noise suppression and baseline correction capability of the implemented EMD have been described in this section. The noisy signal has been filtered using the partial reconstruction algorithm as described earlier. The numerical expression for the noisy sinusoidal signal is given as

$$x(i) = 100 \cdot \sin(2 \cdot \pi \cdot 0.01i) + A \cdot n_g(i) + V_{DC}, \quad (19)$$

where  $0 < i \leq 1024$ ,  $A$  indicates the multiplication factor, and  $n_g(i)$  is the spatially uniformly distributed random white Gaussian noise which is introduced with the sinusoidal signal of 100 kHz frequency. The DC baseline of  $n_g(i)$  is shifted in the negative value  $V_{DC}$  with  $A = 100$ , as shown in Fig. 6. This random white Gaussian noise has been captured from the Tektronix AFG3103C function generator with different voltage amplitudes  $A$ . The signal-to-noise ratio (SNR) of the noisy signal is calculated as  $SNR = 20 \cdot \log_{10}(\max_i(x(i))/\max_i(A n_g(i)))$ ,  $0 < i \leq 1024$ . Figure 7 provides the waveform plots of noisy sinusoidal inputs  $x(t)$  and their EMD filtered outputs  $\tilde{x}_a(t)$ . It can be observed that EMD processed signal  $\tilde{x}_a(t)$  provides the smooth sinusoidal signal with zero DC baseline. The obtained

设置不同大小噪声的对比组，通过对波松相关系数COR，证明了EMD降噪的有效性

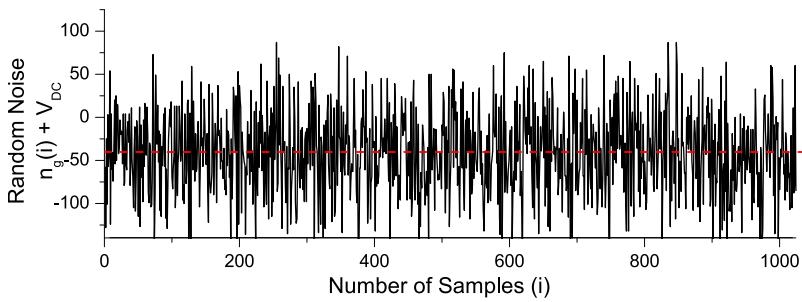


FIG. 6. Random white Gaussian noise  $A \cdot n(t)$  with negative DC baseline.

IMF index  $k_c$  is 1 for the signal reconstruction. The similarity between actual signal  $x_a(t)$  and EMD processed signal  $\tilde{x}_a(t)$  has been analyzed by calculating the Pearson correlation coefficient,<sup>24</sup> as described in (20). The  $\bar{x}$  and  $\bar{y}$  are the mean values of the signal  $x_a(t)$  and  $\tilde{x}_a(t)$ , respectively. It is observed that there is a decrement in COR of the signal due to the increment of noise. However, the COR of 0.99192 was obtained from the noisy signal with the SNR of 6.020 dB, as shown in Fig. 7(h),

$$COR = \frac{\sum_{i=1}^N (x_a(i) - \bar{x})(\tilde{x}_a(i) - \bar{y})}{\sqrt{\sum_{i=1}^N (x_a(i) - \bar{x})^2 \sum_{i=1}^N (\tilde{x}_a(i) - \bar{y})^2}}. \quad (20)$$

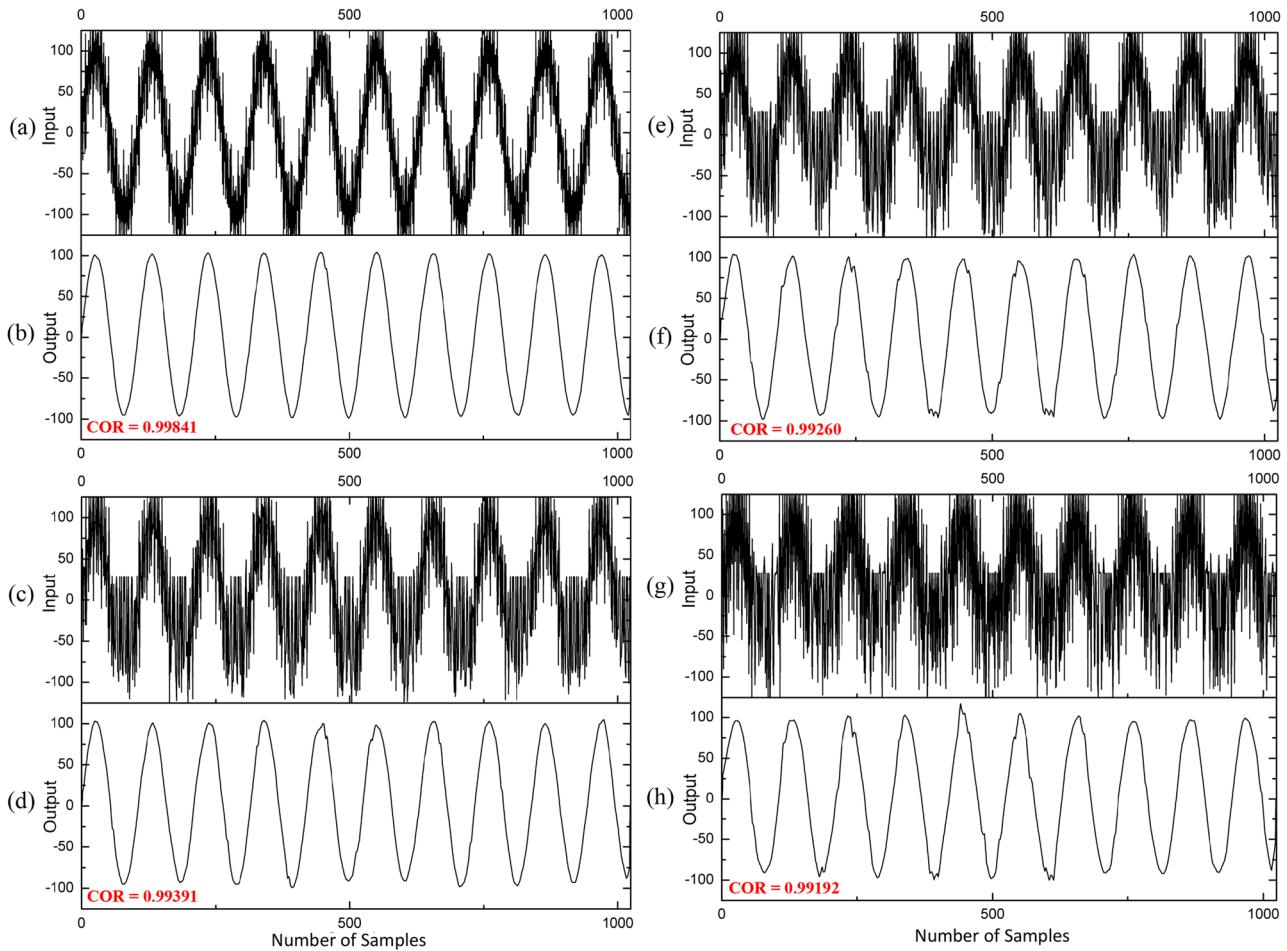


FIG. 7. Baseline correction and noise filtering of sinusoidal noisy signals using the EMD software. Input waveforms are the noisy signals  $x(t)$  with the SNR of (a) 16.48 dB, (c) 12.04 dB, (e) 9.118 dB, and (g) 6.020 dB. Output waveforms are the EMD processed signals  $\tilde{x}_a(t)$  with the COR of (b) 0.99841, (d) 0.99391, (f) 0.99260, and (h) 0.99192.

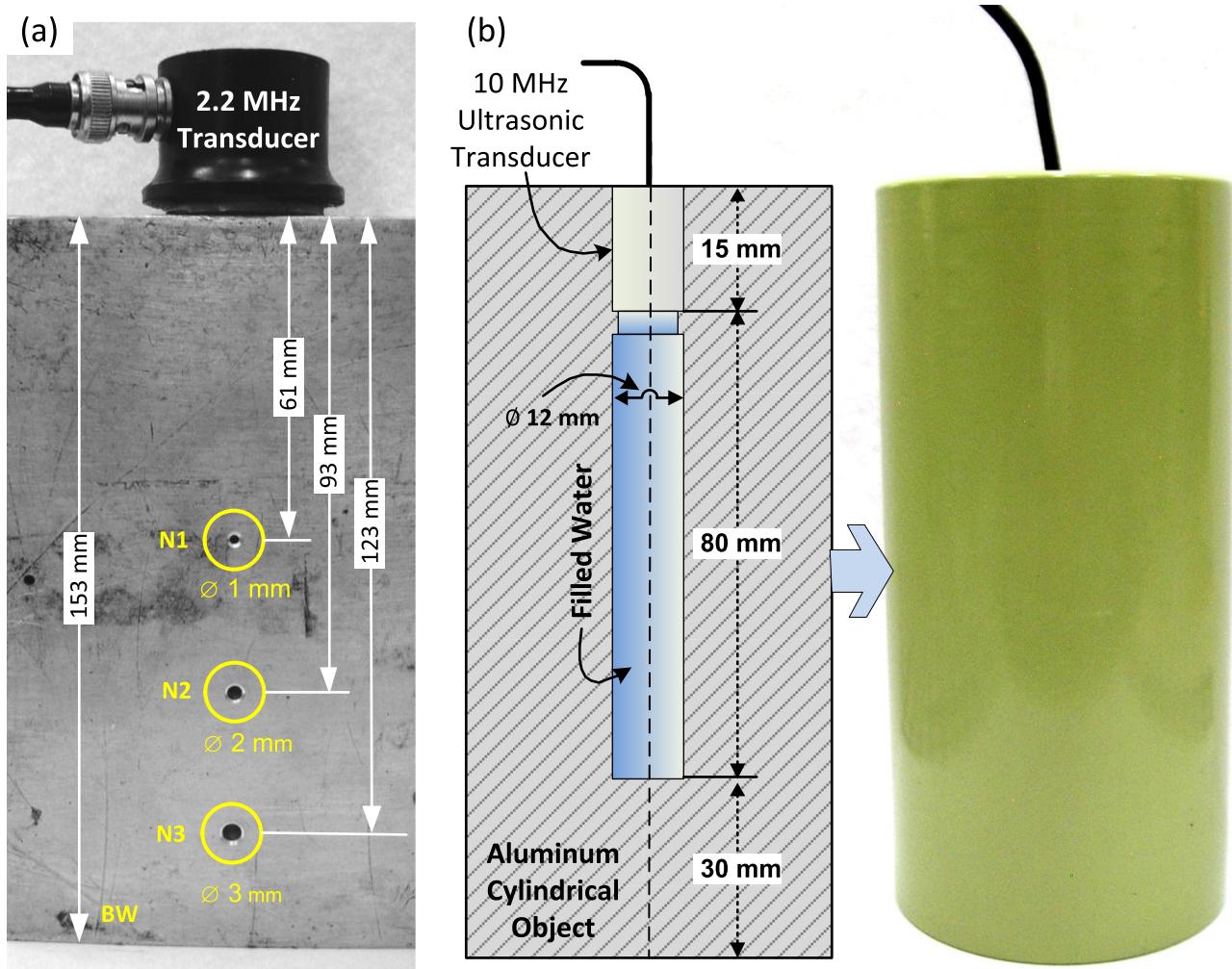


FIG. 8. Test setup for ultrasonic pulse-echo measurements: (a) aluminum object under test with three vertical holes  $N_1$ ,  $N_2$ , and  $N_3$ , respectively, and a 2.2 MHz contact transducer and (b) water filled aluminum test setup and a 10 MHz immersion transducer.

2.2 MHz frequency and 25 mm casing diameter. The sampling rate of 16 MHz with 8-bit resolution is chosen for the data acquisition. The actual signal  $x_a(t)$ , noisy (non-EMD) signal  $x(t)$ , and its EMD processed signal  $\tilde{x}_a(t)$  are captured in GUI

software as shown in Fig. 9. The obtained IMF index  $k_c$  is 2 for the contact ultrasonic testing. The modified SNR function is used to analyze the enhancement of the pulse-echo signal using EMD based signal filtering.<sup>14</sup> The SNR function is

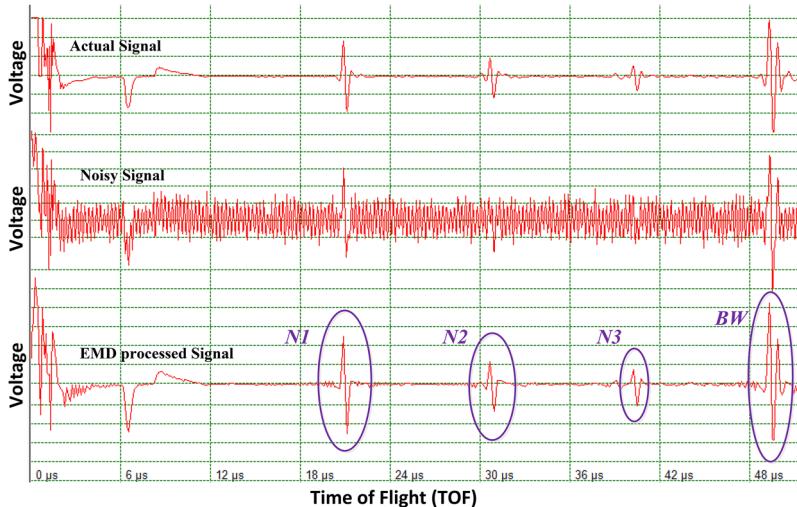
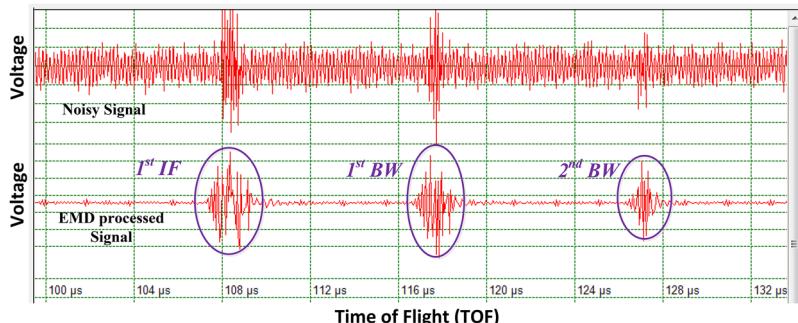


FIG. 9. Screenshot of GUI software for online viewing of actual signal  $x_a(t)$ , noisy signal  $x(t)$ , and EMD processed signal  $\tilde{x}_a(t)$ :  $k_c = 3$ ,  $n = 1024$ , sampling rate  $f_s = 16$  MHz, and transducer frequency  $f_T = 2.2$  MHz.

TABLE II. SNR enhancement of ultrasonic signals.

Echoes	Depth from transducer (mm)	SNR noisy signal $x(t)$	SNR EMD processed signal $\tilde{x}_d(t)$	SNR enhancement
Contact testing				
$N1$	61	-14.2239	+10.3968	+24.6207
$N2$	93	-17.2012	+8.6246	+25.8258
$N3$	123	-18.5110	+3.7801	+22.2911
$BW$	153	-12.5792	+15.1700	+27.7492
Immersion testing				
1st <i>IF</i>	80	-27.1326	-0.8853	+26.2473
1st <i>BW</i>	110	-29.6862	-7.4857	+22.2005
2nd <i>BW</i>	140	-36.0773	-17.1687	+18.9086



## 信噪比函数证明了EMD对信号的增强效果

$$SNR = 10 \cdot \log_{10} \left[ \frac{\sum_{i=T-W/2}^{T+W/2} x^2(i)}{\sum_{i=1}^n x^2(i) - \sum_{i=T-W/2}^{T+W/2} x^2(i)} \right], \quad (21)$$

where  $T$  is the location of flaw/hole and  $W$  is the pulse width of the echo signal. These parameters are obtained manually through visual examination. The same  $SNR$  function is used for noisy input and its EMD processed output. Here, it can be observed that ultrasonic echoes from the three holes  $N1$ ,  $N2$ , and  $N3$  and a back-wall  $BW$  are clearly detected and visualized in the EMD processed output signal with the  $SNR$  enhancement (Table II) of  $>22$  dB.

A 10 MHz, unfocused, un-damped, 6 mm crystal diameter, water immersible transducer is used for the immersion NDT. The aluminum based mechanical cylindrical object has been made in the laboratory, as shown in Fig. 8(b). After the high voltage excitation of the ultrasonic transducer, the first echo (1st *IF*) is detected from the water-aluminum interface, and the second echo (1st *BW*) is detected from the back wall surface of the aluminum block. Figure 10 provides pulse-echo responses from the aluminum block. The obtained IMF index  $k_c$  is 3 for immersion ultrasonic testing. The  $SNR$  improvement of the EMD processed pulse-echo signals are 26.25 dB, 22.20 dB, and 18.91 dB for the signals

1st *IF*, 1st *BW*, and 2nd *BW*, respectively. For the PLI-EMD implementation, the  $SNR$  enhancement of the 1st *BW* echo is 26.43 dB.

## D. Performance of the EMD based denoise system

Table III describes the calculated parameters of the real-time EMD based denoise system. The hardware-accelerated method for EMD data processing has been developed in Ref. 9, but they did not receive the real time results as they require 1.43 s to process 2048 data samples. The method described in Ref. 10 for the ECG system divides the whole dataset into the multiple sections of 1000 data byte samples where each section needs 2.48 s for EMD processing. However, the complete FPGA based implementation of PLI based EMD requires 0.0001 s, as described in Ref. 8. The method in Ref. 13 developed the fast EMD implementation and needs only 0.000075 s for the processing of 1000 samples using PLI-EMD. Our proposed PLI-EMD implementation needs the least processing time of 0.0000491 s, compared to all previously implemented PLI-EMD. The processing time for the implemented CSI-EMD is 0.00150 s. The current EMD implementation has no limitation on memory cost or area. The sampling rate for the current implementation can be increased

FIG. 10. Screenshot of GUI software for online viewing of noisy signal  $x(t)$  and EMD processed signal  $\tilde{x}_d(t)$ :  $k_c = 2$ ,  $n = 1024$ , sampling rate  $f_s = 25$  MHz, and transducer frequency  $f_T = 10$  MHz.

TABLE III. Calculated parameters of EMD based denoise system.

	Envelope generation method	Sampling rate (MHz)	Processing time (s)	Data samples #
Current work	CSI	50	0.001500	1000
Current work	PLI	50	0.0000491	1000

up to 100 MHz. The highest sampling rate for the current system implies the feasibility of the real time ultrasonic measurement systems. The work presented in Ref. 8 filters simply the high-frequency sinusoidal interference from the ultrasonic data. But the current work describes the noise filtering application applying the actual ultrasonic signal which is degraded by the white Gaussian noise.

## VII. CONCLUSION

This paper has presented the implementation of the real time EMD algorithm based signal processing system, especially for ultrasonic pulse-echo NDT applications. The PLI and CSI based EMD application has been developed in the visual software environment. In order to solve the cubic tridiagonal matrix to generate the upper/lower envelope, the TDMA/Thomas algorithm has been applied which reduces the computation cost to  $\sim O(n)$ . The total time complexity has been calculated for both PLI-EMD and CSI-EMD methods, and it is equal to the  $O(n \log n)$ . The fully data-driven partial reconstruction scheme has been adopted for the signal filtering. The EMD software can correct the DC shifted baseline and reduce the noise by extracting the particular IMF elements from the noisy input signal. It is observed that the DC shifted white Gaussian noises are filtered out from the sinusoidal signals using the implemented EMD. The proposed methodology has demonstrated significant improvement of the SNR up to +27 dB for the ultrasonic pulse-echo signals, obtained from the artificial holes using the contact method as well as from the water-immersed aluminum object using immersion method. The proposed EMD implementation has low processing time (PLI-EMD), high sampling rate (PLI-EMD/CSI-EMD), and no limitation on memory cost (PLI-EMD/CSI-EMD).

- <sup>1</sup>V. Matz, R. Smid, S. Starman, and M. Kreidl, *Ultrasonics* **49**, 752 (2009).
- <sup>2</sup>J. Chen, Y. Shi, and S. Shi, *Int. J. Pressure Vessels Piping* **76**, 619 (1999).
- <sup>3</sup>R. Drai, F. Sellidj, M. Khelil, and A. Benchaala, *Ultrasonics* **38**, 503 (2000).
- <sup>4</sup>M. Izquierdo, M. Hernandez, O. Graullera, and L. Ullate, *Ultrasonics* **40**, 259 (2002).
- <sup>5</sup>J. C. Lazaro, in *Proceedings of IEEE Ultrasonics Symposium, 2002* (IEEE, 2002), Vol. 1, pp. 777–780.
- <sup>6</sup>E. Pardo, J. S. Emeterio, M. Rodriguez, and A. Ramos, *Ultrasonics* **44**, e1063 (2006).
- <sup>7</sup>N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, *Proc. R. Soc. London, Ser. A* **454**, 903 (1998).
- <sup>8</sup>Y. Y. Hong and Y. Q. Bao, *IEEE Trans. Instrum. Meas.* **61**, 3175 (2012).
- <sup>9</sup>L. Wang, M. I. Vai, P. U. Mak, and C. I. Leong, in *Proceedings of 3rd International Conference Biomedical Engineering Informatics* (IEEE, 2010), Vol. 2, pp. 912–915.
- <sup>10</sup>M. H. Lee, K. K. Shyu, P. L. Lee, C. M. Huang, and Y. J. Chiu, *IEEE Trans. Ind. Electron.* **58**, 2473 (2011).
- <sup>11</sup>S. Cagdas and A. Celebi, in *Proceedings of IEEE Signal Processing and Communications Applications Conference* (IEEE, 2012), pp. 1–4.
- <sup>12</sup>S. Shukla, S. Mishra, B. Singh, and S. Kumar, *IEEE Trans. Ind. Appl.* **53**, 2392 (2017).
- <sup>13</sup>P. Y. Chen, Y. C. Lai, and J. Y. Zheng, *IEEE Trans. Ind. Electron.* **63**, 3686 (2016).
- <sup>14</sup>T.-L. Chen, P.-W. Que, Q. Zhang, and Q.-K. Liu, *Rev. Sci. Instrum.* **76**, 085109 (2005).
- <sup>15</sup>Y. Mao and P. Que, *Russ. J. Nondestr. Test.* **43**, 196 (2007).
- <sup>16</sup>A. O. Boudraa and J. C. Cexus, *IEEE Trans. Instrum. Meas.* **56**, 2196 (2007).
- <sup>17</sup>L. H. Thomas, “Elliptic problems in linear differential equations over a network,” Watson Sc. Comp. Lab. Rep., Columbia University, New York, (1949).
- <sup>18</sup>P. Flandrin, G. Rilling, and P. Goncalves, *IEEE Signal Process. Lett.* **11**, 112 (2004).
- <sup>19</sup>N. E. Huang, M.-L. C. Wu, S. R. Long, S. S. Shen, W. Qu, P. Gloersen, and K. L. Fan, *Proc. R. Soc. A* **459**, 2317 (2003).
- <sup>20</sup>Z. Wu and N. E. Huang, *Proc. R. Soc. A* **460**, 1597 (2004).
- <sup>21</sup>Y.-H. Wang, C.-H. Yeh, H.-W. V. Young, K. Hu, and M.-T. Lo, *Physica A* **400**, 159 (2014).
- <sup>22</sup>Z. Wu and N. E. Huang, *Adv. Adapt. Data Anal.* **01**, 1 (2009).
- <sup>23</sup>Z. Wu and N. E. Huang, *Adv. Adapt. Data Anal.* **02**, 397 (2010).
- <sup>24</sup>J. Chen, Y. Huang, J. Benesty, and I. Cohen, *Noise Reduction in Speech Processing*, Springer Topics in Signal Processing (Springer, 2009).