IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 October 2022, accepted 25 October 2022, date of publication 31 October 2022, date of current version 4 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3218417

RESEARCH ARTICLE

On the Memory Cost of EMD Algorithm

HSU-WEN VINCENT YOUNG[1], YU-CHUAN LIN[2], AND YUNG-HUNG WANG[2]
[1]Department of Electronic Engineering, Chung Yuan Christian University, Taoyuan 320314, Taiwan
[2]Department of Mechanical Engineering, National Central University, Taoyuan 32001, Taiwan

Corresponding author: Yung-Hung Wang (yunghung@ncu.edu.tw)

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 111-2634-F-006-003- and Grant 111-2221-E-008-047-MY2.

ABSTRACT Empirical mode decomposition (EMD) and its variants are adaptive algorithms that decompose a time series into a few oscillation components called intrinsic mode functions (IMFs). They are powerful signal processing tools and have been successfully applied in many applications. Previous research shows that EMD is an efficient algorithm with computational complexity $O(n)$ for a given number of IMFs, where n is the signal length, but its memory is as large as $(13 + m_{imf})\, n$, where m_{imf} is the number of IMFs. This huge memory requirement hinders many applications of EMD. A physical or physiological oscillation (PO) mode often consists of a single IMF or the sum of several adjacent IMFs. Let m_{out} denote the number of PO modes and, by definition, $m_{Out} \leq m_{imf}$. In this paper, we will propose a low memory cost implementation of EMD and prove that the memory can be optimized to $(2 + m_{out})\, n$ without aggravating the computational complexity, while gives the same results. Finally, we discuss the optimized memory requirements for different noise-assisted EMD algorithms.

INDEX TERMS EMD, EEMD, CEEMD, memory cost.

I. INTRODUCTION

EMD [1] is a relatively recent algorithm of time series analysis that is nonlinear and adaptive in nature. The results of applying EMD to time series data in general is the decomposition of a signal into multiple components, known as IMFs, through a featured sifting process. The IMFs are extracted at different time scales, ranging from small to large, by iterating the sifting process. In each iteration, the local maxima (resp. minima) are detected and then connected using cubic spline interpolation (CSI), which is the most commonly adopted approach for EMD [2], thus forming the upper (resp. lower) envelope. Then, the mean of the upper and lower envelopes, which is in general a relatively slow oscillating signal, is subtracted from the previous input, known as a prototype (proto-) IMF, to get a new proto-IMF. When the sifting has been repeated for a number that is a priori set by the researcher, the resulting series will be deemed an IMF and is then subtracted from the original signal, whose result is in turn used as the input signal for the next stage of IMF extraction.

The associate editor coordinating the review of this manuscript and approving it for publication was Olutayo O. Oyerinde.

EMD sometimes suffers from the phenomenon known as mode-mixing [3], [4], [5], [6], [7], [8], the later developed disturbance-assisted EMD algorithms (DA-EMD) such as EEMD [4], CEEMD [5], masking EMD [6], UPEMD [7], and MEMD [8] that contain the original EMD as the kernel of the algorithms, can be applied to overcome this problem. The DA-EMD algorithms have been successfully applied in diverse fields including physics [1], mechanical system [9], image analysis [10], biomedicine [11], and electrical engineering [12], to name a few.

EMD and its variants have many applications. First, an IMF or the sum of several adjacent IMFs often corresponds to a particular physical/physiological mechanism/source [1], [7], [10]. In addition, the IMFs can also be used as the inputs to a neural-network [13]. Finally, EMD can serve as an adaptive filter in which case the sum of selected IMFs can be taken as the denoised signal [12].

Computation complexity and memory cost are two key indices to measure the efficiency of EMD [14], [15], [16]. Previous research indicates that the EMD is a computation efficient algorithm [17] with complexity $O(n)$ for a given number of IMFs [18]. However, it was shown that its memory

cost is $M = (13 + m_{imf})\, n$, where n is the data length, and m_{imf} is the number of IMFs [18]. Although it is linear [19], but with a coefficient as large as $13 + m_{imf}$.

Suppose that EMD is applied as a bandpass filter and the input signal is decomposed into 7 IMFs, i.e., $m_{imf} = 7$. Then the total factor is 20. Such a huge memory requirement will hamper many applications of EMD. For instance, it remains a challenge to implement real-time DA-EMD via a pure software approach in a computation and memory resource limited micro-controller (MCU) [20], [21]. Consider implementing EMD on a 32 bit MCU in a wearable device, where EMD is applied to a 30-s electroencephalography (EEG) signal sampled at 250Hz, i.e., $n = 7,500$. The memory requirement for applying EMD as a single bandpass filter is 600 kB, which is far beyond the memory limit of an MCU [22].

Moreover, when EMD is applied to GPU parallel computations, large memory demands may lead to excessive memory allocations and transfer, which may dramatically increase the run-time [23], [24], [25].

We propose in this article a low memory cost EMD implementation called low-memory EMD (LMEMD) in order to optimize the memory while giving the identical results as the original EMD. According to [18], the memory cost of EMD consists of three parts.

part 1). The output IMFs requires $m_{imf}\, n$ (float).

A physical or physiological oscillation (PO) mode often consists of a single IMF or the sum of several adjacent IMFs. Let m_{out} denote the number of PO modes and $m_{Out} \le m_{imf}$. Then this part of memory can be reduced to $m_{out}\, n$.

part 2). The CSI procedure requires $10n$ (float).

To reduce the needed memory storage, the entire signal is often partitioned into windows of fixed length. Each side of the window is then extended by a fixed overlap length to reduce the boundary error and to ensure the continuity between adjacent time windows [18], [20], [26], [27], [28], [29]. Then EMD is executed in each window sequentially from left to right.

However, these approaches still have a few drawbacks. First, the memory cost is never investigated in detail. Second, they invoke some empirical criteria for the overlap lengths. Moreover, the discarded data might introduce boundary errors [30] that will propagate into interior domain, but the error of an IMF has never been precisely analyzed because the mathematical theory of EMD is far from complete [31]. If the overlap length is too short, the boundary error effect will be large. If the overlap length is too long, it will increase the memory usage.

It is recently proved that the boundary error caused by the window-wise approach decays exponentially fast with the number of local maxima (resp. minima) in the overlap region [32]. Let n_L, n_R, and n_C denote the number of local maxima (resp. minima) in the left overlap, right overlap, and central region, respectively in Fig. 1(a), and their values are (window) frame independent. We will prove that as n_R and n_L exceed a certain number, the errors of the CSI and any IMF will achieve machine zero, and the required memory can be reduced to $8n_w$. Since the memory in the CSI is proportional to the window size $n_w (= n_L + n_c + n_R)$, if we opt to choose $n_w \ll n$, the memory cost in the CSI procedure becomes negligible.

part 3). The input signal and working memory require $3n$ (float).

We will apply an alternative-envelope-sliding-window (AESW) approach as illustrated in Fig. 1(b) so that the signals before and after a sifting iteration can share the same memory to reduce the memory to $2n$.

The paper is organized as follows. Section II summarizes the concepts and algorithms of CSI and EMD. Section III proposes the method of LMEMD and proves the above statements concerning its memory cost. In Section IV, we present the experimental results of the LMEMD on a personal computer. Section V concludes the paper with a brief discussion.

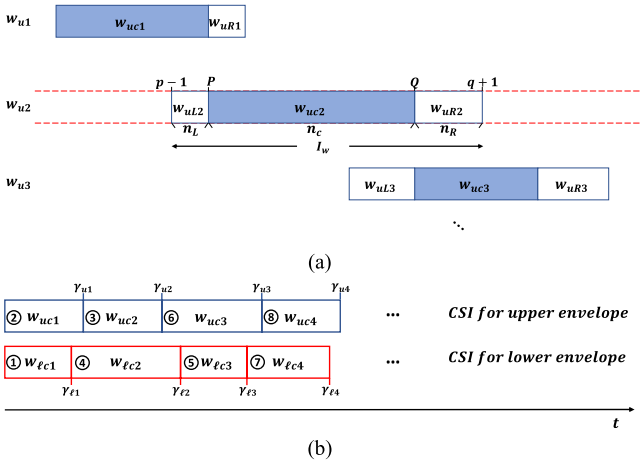

FIGURE 1. Schematic of the sliding window for LMEMD. (a) $u(t)$; (b) The AESW approach in LMEMD. The windows are processed in sequential order ①, ②, ③ ⋯.

II. EMD BASICS

A. CUBIC SPLINE

Given an input signal $x(t) \in R^n$, $t = 1, 2, \cdots, n$. Let $[\gamma_k]$, $1 \le k \le K$ denote the position of the k-th local maximum of $x(t)$ and $\Gamma_k = x(\gamma_k)$, the local maximum value. The entire time duration of the processed signal, I, is decomposed into sub-intervals: $I_k[\gamma_k \gamma_{k+1})$, $1 \le k < K$ without overlapping. Let $h_k = \gamma_{k+1} - \gamma_k$ denote the scale length.

The upper envelope $u(t)$ is constructed piecewise using a series of (piecewise) cubic polynomials $u_k(t)$ that enforce the continuities of u, $u'(t)$ and $y_k \triangleq u''(t)$ at the local maxima [33], [18]. The vector composed by second derivatives of $u_k(t; \gamma_k, \Gamma_k)$ is then given by the following tridiagonal system of linear equations:

$$\mathbf{A}\vec{y} = \vec{d},$$

where

$$\mathbf{A} = \begin{bmatrix} b_2 & c_2 & & & & & \\ a_3 & b_3 & c_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_k & b_k & c_k & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{K-2} & b_{K-2} & c_{K-2} \\ & & & & & a_{K-1} & b_{K-1} \end{bmatrix},$$

$$a_k = h_{k-1}, \quad b_k = 2(h_{k-1} + h_k), \; c_k = h_k,$$

$$d_k = 6\left[\left(\Gamma_{k+1} - \Gamma_k\right)/h_k - \left(\Gamma_k - \Gamma_{k-1}\right)/h_{k-1}\right] \quad (1)$$

In particular, note that the size of the matrix A is $(k-2) \times (k-2)$. Natural boundary conditions $y_1 = y_K = 0$ have been imposed in (1). Let $\tau = t - \gamma_k$ for t sitting in I_k. After $\vec{y}$ is found numerically, the functional dependencies of $u_k(t)$ can be written as follows [32]:

$$\begin{aligned} u_k(t) &= u_k\left(t - \gamma_k; y_k, y_{k+1}, x(\gamma_k), x(\gamma_{k+1})\right) \\ &= u_k\left(\tau; y_k, y_{k+1}, x(\gamma_k), x(\gamma_{k+1})\right). \end{aligned} \quad (2)$$

The lower envelope $l(t)$ can be constructed in the same way.

B. EMD ALGORITHM

EMD consists of a main procedure with multiple stages (the actual number of needed stages is determined by the specific data to be processed) in which a particularly important operation called sifting (function SIFT) is iterated for several times at every stage except the last one. Let $x(t)$ and $\tilde{x}(t)$ be the proto-IMF before and after the sifting iteration. The sifting iteration is as follows.

$$\tilde{x}(t) = x(t) - \left[u(t; x(t)) + l(t; x(t))\right]/2 \quad (3)$$

Sifting Procedure: $\tilde{x}(t) = \mathrm{SIFT}(x(t))$

1: Identify the local maxima and the local minima.
2: Solve for the maxima $\vec{y}$ and compute $u(t)$
3: Solve for the minima $\vec{y}$ and compute $l(t)$
4: Obtain the proto-IMF $\tilde{x}(t)$ by (3).

In the original EMD algorithm, there is nothing like step 5 above, and, instead step 6 here is what is to be executed. The details of step 5 will explained soon in the next section, and this step is of some novelty and of particular importance in our realization and applications of LMEMD.

III. THE LMEMD ALGORITHM

A. OPTIMIZATION OF THE OUTPUT MEMORY

A physical/physiological oscillation (PO) mode may either be represented by a single IMF or by the sum of several adjacent IMFs. Let m_{out} denote the number of PO modes and thus $m_{Out} \leq m_{imf}$. Suppose that only the PO modes are really required for the specific scientific/engineering application at hand, then it is not necessary to store all the IMFs simultaneously, thus allowing a way to reduce the output memory. The algorithm is listed above in our version of the main procedure of the EMD algorithm in which step 5 instead of

EMD: Main Procedur

Input: signal $x(t)$, number of IMFs m_{IMF}, and number of siftings n_s.
Output: all IMFs $C_m(t), m = 1 : m_{IMF}$ or all PO modes $PO_m(t), m = 1 : m_{out}$.

1: Set residue $R_0(t) = x(t)$.
2: for $m = 1 : m_{imf} - 1$
3: Set proto-IMF $x(t) = R(t)$.
4: for $s = 1 : n_s$
Apply one sifting to obtain $\tilde{x}(t) = \mathrm{SIFT}(x(t))$.
if (apply LMEMD)
5: if $(m \in PO_k)$ Set $PO_k = PO_k + C_m(t)$.
else (apply original EMD)
6: Output $C_m(t)$
7: Compute residue $R_m(t) = x(t) - C_m(t)$.

step 6 is executed. For an example, if EMD is applied as a bandpass filter [34], then $m_{out} = 1$. An example will be given in Section IV.

B. OPTIMIZATION OF THE CSI MEMORY

In order not to repeat essentially identical derivations, here we choose to analyze the properties of the upper envelope, and those similar assertions for the lower envelope can be obtained in a completely analogous way.

1) BOUNDARY ERROR AND LOCAL PROPERTY OF THE RESPONSE

In the following, y will stand for the response vector/function and let δy denote the referred boundary error whose detailed definition is given in the following.

We first analyze the local property of $\vec{y}$ for the upper envelope using the entire domain $t \in I$. Denote by V the inverse of the matrix A in (1), writing out its columns, $V[\vec{v}_2, \cdots \vec{v}_{K-1}][v_{i,j}]$, where $\vec{v}_j = [v_{2,j}, \cdots, v_{K-1,j}]^T$. Let $\vec{e}_j$ be the unit vector with a one in the j-th entry and zero elsewhere, i.e., $\vec{e}_j = [\cdots, 0, 1, 0 \cdots]^T$. The solution $\vec{v}_j$ due to an impulse vector $\vec{e}_j$ satisfies the equations: $\vec{v}_j = V\vec{e}_j$. Therefore, $v_{i,j}$ can be interpreted as the response at the i-th knot (position of a local maximum) due to an impulse at j. After $v_{i,j}$ has been obtained, the solution $\vec{y}$ can be expressed as the linear combination of $\vec{v}_j$ as follows.

$$\vec{y} = d_2\vec{v}_2 + d_3\vec{v}_3 + \cdots + d_{K-1}\vec{v}_{K-1} \quad (4)$$

Let $r_{i,j}$ denote the decay rate of y_i at i due to an impulse at j, and ξ_i denote the scale ratio at i.

$$r_{i,j}\frac{v_{i+1,\mathrm{j}}}{v_{i,j}}, \quad 0 < \xi_i\frac{h_{i-1}}{h_i} = \frac{\gamma_i - \gamma_{i-1}}{\gamma_{i+1} - \gamma_i} \quad (5)$$

Theorem 1: The $r_{i,j}$ are localized and "causal", meaning that: for $i < j$,

(a): $r_{2,j} = -2(1 + \xi_2) < -2$,

$$r_{i,j} = -2\left[1 + \left(1 + \frac{1}{2r_{i-1,j}}\right)\xi_i\right] < -2, \quad i > 2. \quad (6)$$

(b): $r_{i,j}$ is independent of position j.

The proof of statement 1(a) is given in [32], and 1(b) directly follows from 1(a). It indicates that $v_{i,j}$ decays exponentially with distance $|i-j|$ with decay rate $|r_{i,j}| > 2$.

It implies that $\vec{v}_j$ and $\vec{y}_j$ are very localized in the sense that the impulse at j mainly affect its nearby local maxima.

We next investigate the local property of δy using a segment of data $t \in w_{ui} = [\gamma_{p-1}\gamma_{q+1}]$ as shown in Fig. 1. Let n_w, n_C, n_L and n_R denote the numbers of local maxima points in w_{ui}, w_{uci}, w_{uLi} and w_{uRi} (the definitions of these intervals are demonstrated in Fig. 1), respectively, and n_C is a fixed number. For the sake of convenience, let us use the following convention: if a f is a function that depends on the entirety of the input data, i.e., $t \in I$, we then let the function f^* with superscript "$*$" denote the "truncated" version of the function that only depends on $t \in w_{ui}$, which is obtained by throwing away the input data outside this particular interval but still enforcing the interpolation and continuity requirements. Also, let $\delta f \triangleq f^* - f$ denote the error caused by discarding boundary data. Also let $q+1$ and Q denote the indices of last local maxima in w_{uRi} and w_{uci}. WLOG, set $n_L = n_R$, then

$$n_R q - Q + 1, \quad \text{and } n_w = 2n_R + n_C \tag{7}$$

Theorem 2: There exists a number ρ, which depends on the specific signal being processed, such that if $n_R \geq \rho$, the boundary error δy_k in the central window, w_{uci}, will be zero in a processor

We will prove Theorem 2 in Theorem 5 and Theorem 7 later. The objective is to establish Theorem 2 and to find the value of ρ as small as possible in order to reduce the size of the overlap window w_{uLi} and w_{uRi}.

Following [32], since $x(t) = x^*(t)$ in $t \in w_{ui}$, it leads to $\delta d_k = d_k^* - d_k = 0$ for $p \leq k \leq q$. Then the equations of $\delta\vec{y}\vec{y}^* - \vec{y}$ are written as

$$\begin{bmatrix} a_p & b_p & c_p & & & & \\ & a_{p+1} & b_{p+1} & c_{p+1} & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & a_{q-1} & b_{q-1} & c_{q-1} & \\ & & & & a_q & b_q & c_q \end{bmatrix} \begin{pmatrix} \delta y_{p-1} \\ \delta y_p \\ \cdot \\ \cdot \\ \cdot \\ \delta y_q \\ \delta y_{q+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \tag{8}$$

Let $\mathbf{A_w}$ be a submatrix of matrix $\mathbf{A}$ in (1) formed by taking the rows and columns ranging from p to q. Move the terms $a_p\delta y_{p-1}$ and $c_q\delta y_{q+1}$ in the first and last equation from LHS to RHS. Following the same procedure in deriving (4), it yields

$$\begin{aligned} \mathbf{A_w}\delta\vec{y} &= \beta_p\vec{e}_p + \beta_q\vec{e}_q, \\ \delta\vec{y} &= \beta_p\vec{v}_p + \beta_q\vec{v}_q, \quad \beta_q = -h_q\delta y_{q+1} = -h_q y_{q+1} \end{aligned} \tag{9}$$

Vector $\delta\vec{y}$ is obtained by the linear combination of $\vec{v}_p$ and $\vec{v}_q$. Without loss of generality (WLOG), consider $\delta\vec{y} = \beta_q\vec{v}_q$. The last equality in (9) holds because of the BC, $y_{p-1}^* = 0$. Note that y_{q+1} on the RHS of (9) remains unknown and must be solved numerically. Let $\bar{k} = q - k > 0$ as the distance between local maximum with index k and impulse position q. Define the average decay rate $\bar{r}$ from $q-1$ to k, recalling the definition as in (5), by

$$\bar{r}^{\bar{k}} r_{k,q} r_{k+1,q} \cdots r_{q-1,q}, \quad \text{where } \bar{k} = q - k \tag{10}$$

Lemma 3:

$$\begin{aligned} v_{k,q} &= \bar{r}^{-\bar{k}} v_{q,q} \text{ and} \\ \delta y_k &= v_{k,q} h_q y_{q+1} < 2^{-\bar{k}} \left| v_{q,q} h_q y_{q+1} \right|. \end{aligned}$$

Proof: The decay rate $r_{i,j}$ is obtained by recursively applying (6) from $i = p$ to q. Next, $v_{q,q}$ is obtained by (1) with index q and $d_q = 1$ as follows:

$$v_{q,q} = (a_q / r_{q-1} + b_q)^{-1} \tag{11}$$

Finally, $v_{k,q}$ is recursively obtained backward from $i = q-1$ to k using (5) and (10). ■

An example of the curve δy_k versus local maximum index k are plotted on the red curve in Fig. 2, which shows that $|\delta y_k|$ decays exponentially with the distance to the right boundary $|q+1-k|$.

2) EFFECT OF BOUNDARY ERROR δy ON ρ

Definition 4: Machine zero (0). If a number z is identified and treated as zero in a processor, i.e., $z \fallingdotseq 0$, then z is called machine zero. Similarly, if two numbers z_1 and z_2 are identical in processor, then the difference between these two numbers is also called machine zero, i.e., $z_2 - z_1 \fallingdotseq 0$ or $z_1 \fallingdotseq z_2$.

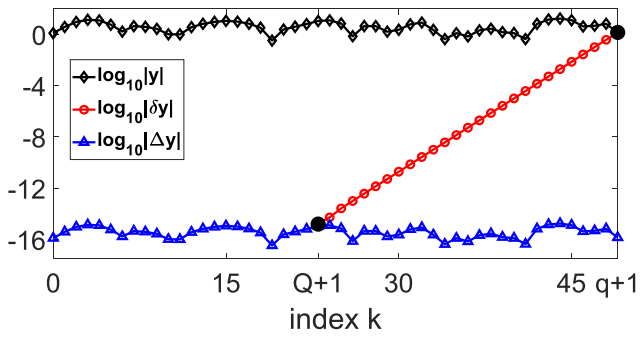

FIGURE 2. Estimation of ρ for the white noise in Example 2. In this example δy and Δy intersects at point $Q+1$ so that ρ is estimated from the intersection point. Based on the figure, it yields $\rho = q - Q + 1 = 48 - 21 + 1 = 28$. Note that on the left of $Q+1$, $\log_{10}(|\delta y| = 0) \to -\infty$ and cannot be displayed on the figure.

A real number z in a processor can be represented by the scientific notation, $z = \mu \cdot 10^E$ with $1 \leq |\mu| < 10$, $E \in \mathbb{Z}$, $E_1 \leq E \leq E_2$, and μ has D bit resolution. WLOG, we only consider a 64-bit floating data type. Following the IEEE 754 standard,

$$E_1 = -308, \quad E_2 = 308, \text{ and } D = 52. \tag{12}$$

Thus if $|z| < |\mu| \cdot 10^{E_1}$, then $z0$; on the other hand, if $|z| > |\mu| \cdot 10^{E_2}$, it is interpreted as infinity.

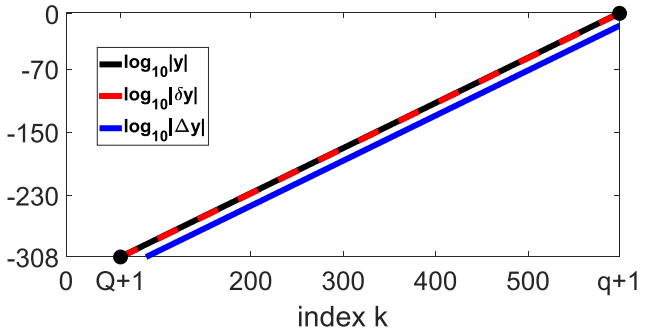

FIGURE 3. Estimation of ρ for the impulse-like signal in Example 1. In this example δy and Δy never intersect so that ρ is estimated using Theorem 5 and is obtained by the point with $|\delta y| = 10^{E_1}$. Based on the figure, it yields $\rho = q - Q + 1 = 598 - 59 + 1 = 541$.

Note that alternatively, E_1 can be set to be -324 at the price of reducing the resolution D to one. The drawback of this definition is that D becomes a variable as $E < -308$. To facilitate the analysis, we will adopt the definition given by (12).

Theorem 5: If $\rho \geq \left\lceil \frac{E_2 - E_1}{\log_{10} 2} \right\rceil = 2047$, then the error in IMF$m$, $\delta C_m(t) \coloneqq 0, t \in w_{uci}$.

Proof: Consider the curve δy_k versus distance $\bar{k}$ in (10). Since $|\delta y_k|$ decays with the distance $\bar{k}$ (Lemma 3), as $\bar{k}$ continuously increases, eventually, $|\delta y_k|$ will intersect the horizontal line 10^{E_1} at a point $Q+1$ so that $|\delta y_Q| \coloneqq 0$ (see Fig. 3). Since $\max |\delta y_{q+1}| = 10^{E_2}$ and $\min |\delta y_Q| = 10^{E_1}$, from Lemma 3 with $k = Q$, it yields

$$|\bar{r}|^{n_R} \geq 10^{E_2 - E_1} \text{ or}$$
$$\rho = \left\lceil \frac{E_2 - E_1}{\log_{10} |\bar{r}|} \right\rceil \geq \left\lceil \frac{E_2 - E_1}{\log_{10} 2} \right\rceil = 2047. \quad (13)$$

By (2), we then have $\delta u(t) = \delta l(t) \coloneqq 0$. Similarly, from (3), we have $\delta \tilde{x}(t)\, 0$. Based on the recursive nature of EMD, it is then clear that $\delta C_m(t) \coloneqq 0$.

3) COMPETITION OF δy AND ROUND-OFF ERROR OF y

In practice, Theorem 5 is often too strict for estimating the upper bound of ρ since it assumes that $|\delta y|$ ranges between $O(10^{-308})$ to $O(10^{308})$. Also, the round-off error in a processor has to be considered.

Definition 6: round-off error Δz. Let $\tilde{z}$ denote the value of z that is being processed in the machine, and thus it is exposed to the round-off error, which can in turn be defined as the difference between the two versions:

$$\Delta z \triangleq \tilde{z} - z. \quad (14)$$

On the other hand, we can express

$$\tilde{z} = (1 + \varepsilon) z, \quad |\varepsilon| = 2^{-D}. \quad (15)$$

Combining (14) and (15), it yields

$$|\Delta z| = \varepsilon |z|. \quad (16)$$

It implies that if $|\varepsilon| < 2^{-D}$, then $\tilde{z}$ and z are essentially indistinguishable, i.e., $\Delta z \coloneqq 0$.

Theorem 7: If $|\delta y_k| \leq |\Delta y_k| = \varepsilon |y_Q|$, and curves $|\Delta y_k|$ and $|\delta y_k|$ intersects at a point $Q+1$, then

(a): $\rho = \frac{D + \log_2 \left| \frac{y_{q+1}}{y_Q} \right| - \log_2 |r_q|}{\log_2 |\bar{r}|} + 1.$

(b): $\rho < \left\lceil \frac{D-1}{\log_2 |\bar{r}|} \right\rceil + 1 \leq 52$ if $y_{q+1} = y_Q$ and $|r_q| \to 2^+$.

Proof: (a) If $|\delta y_Q| \leq |\Delta y_Q|$, then the error is determined by Δy_Q, which in fact makes $\delta y_Q 0$. Substitute $|\delta y_Q|$ into Lemma 3, it yields

$$|\delta y_Q| = |\bar{r}|^{-(\rho-1)} \left| (a_q / r_{q-1} + b_q)^{-1} h_q y_{q+1} \right|$$
$$\leq 2^{-D} |y_Q|, \quad (17)$$

where $\delta y_{q+1} = y_{q+1}$ has been applied because of BC, and $|\Delta y_Q| = \varepsilon |y_Q|, \varepsilon = 2^{-D}$. The value of ρ is then obtained by solving the equation: $|\delta y_Q| = |\Delta y_Q|$. After some algebraic manipulation, it yields

$$\rho = \frac{D + \log_2 \left| \frac{y_{q+1}}{y_Q} \right| - \log_2 \left| \left[\frac{\xi_q}{r_{q-1}} + 2(\xi_q + 1) \right] \right|}{\log_2 |\bar{r}|} + 1 \quad (18)$$

Following (6), the term in the bracket in the numerator can be simplified as follow.

$$\left[\frac{\xi_q}{r_{q-1}} + 2(\xi_q + 1) \right] = r_q. \quad (19)$$

Consequently, ρ can be simplified to what appears in part (a).

(b): From Theorem 1(a), $|r_q| > 2$, then

$$\rho \leq \frac{D + \log_2 \left| \frac{y_{q+1}}{y_Q} \right| - 1}{\log_2 |\bar{r}|} + 1 \quad (20)$$

Part (b) is obtained by substituting $y_{q+1} = y_Q$ into the above equation. ■

The values of the upper bound of ρ estimated using Theorem 5 and Theorem 7(b) differ by approximately 40 times. Next, we will discuss when we should apply Theorem 5.

Theorem 8: Suppose that $x(t)$ is an impulse-like signal which satisfies that $x''(\gamma_k) = 0$ (distributed along a straight line) in $[\gamma_{D1}, \gamma_{D2})$, where $\gamma_{D2} < q$ and $\gamma_{D1} > p$ (Fig. 1(a)), and $x''(\gamma_k) \neq 0$ at $k = D2 + 1$. A schematic of $x(t)$ is shown in Fig. 4.

(a) $|y_k|$ decays exponentially fast with distance $|k - D2|$ at the same decay rate as that of $|\delta y_k|$ in $[\gamma_{D1}, \gamma_{D2}]$.

(b) If $D2 = q$ (see Fig. 1(a)), then the values Δy_k and δy_k will never coincide. Consequently, Theorem 5 should be applied to estimate ρ.

Proof: (a): The second derivative of $x(\gamma_k)$ is approximated by the following difference equation:

$$x''(\gamma_k) = \frac{2}{h_k + h_{k-1}} \left[\frac{x(\gamma_{k+1}) - x(\gamma_k)}{h_k} - \frac{x(\gamma_k) - x(\gamma_{k-1})}{h_{k-1}} \right] \quad (21)$$

In fact, d_k on the RHS of (1) is proportional to the numerator of $x''(\gamma_k)$ in (21). Therefore $d_k = 0$ for $\gamma_{D1} \leq \gamma_k < \gamma_{D2-1}$, and the only non-zero term occurs at d_{D2}. Following (4), it yields

$$\vec{y} = 0\vec{v}_{D1} + \cdots + 0\vec{v}_{D2-1} + d_{D2}\vec{v}_{D2} = d_{D2}\vec{v}_{D2} \quad (22)$$

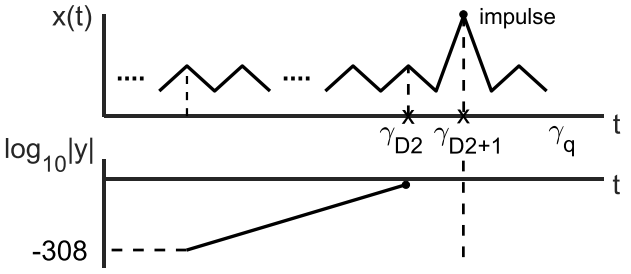

FIGURE 4. Schematic of effect of impulse-like signal. top: impulse-like signal, bottom: decay rate $\log_{10}|y_k|$. versus index k.

Following the procedure in deriving (9), the equations for y for $k = D1$ to $D2$ can expressed in (1) with index $D1 \leq k \leq D2$, and

$$\vec{d} = [0, 0, \cdots 0, d_{D2} - c_{D2} d_{D2+1}]^T. \tag{23}$$

This equation is similar to (9) except that here δy_k is replaced by y_k, so that Theorem 1 still holds for y_k and it decays exponentially with the distance $|k - D2|$. From Theorem 1, the decay rate only depends on the distribution of the locations of the local maxima and is independent of the impulse position d_{D2}. Consequently, $|\delta y_k|$ and $|y_k|$ have the same decay rate. The magnitude of the impulse at the last entry of (23) and β_q in (9) might be different, and therefore the values for δy_k and y_k might be different.

(b) Set $D2 = q$. Following (16), the curves of y_k and Δy_k are parallel, so that the values of δy_k and Δy_k will never be equal. ■

Now we discuss when the condition of Theorem 7 will hold. Let signal $x(t)$ be an amplitude modulated (AM) signal. It is often true for a real-world signal $x(t)$ that it consists of multiple IMFs; otherwise there is no need to apply EMD to decompose the signal. The situation can become very complicated since the contributions from d_k and v_k in (4) need to be taken into account. Based on the local nature of y_k (Theorem 1), its value can be estimated by $x''(\gamma)$ around the neighboring local maximum of γ_k if the order of magnitude of $|x''(\gamma_k)|$ (or d_k) does not vary significantly in the time domain.

Example 1: We investigate a special case where the conditions of Theorem 7 fail. A signal $x(t)$ with $h_k = 1, \forall k$ and $\vec{x}(\gamma) = \vec{e}_{q+1}$. According to Theorem 8, ρ should be estimated by Theorem 5 and is $\rho = 541$ based on Fig. 3.

Example 2: We investigate a case where the condition of Theorem 7 holds. The signal $x(t)$ is a white noise. Fig. 2 plots y and $|\delta y|$ versus index k. All the variables are plotted in log 10 scale. The curves $\log_{10}|y|$, $\log_{10}(|\Delta y|)$, and $\log_{10}|\delta y|$, are plotted in black, blue, and red, respectively. It is observed that the curves $|\delta y|$ and $|\Delta y|$ intersect, thus Theorem 7 holds. It is also observed that $|y|$ ranges approximately from 0.1 to 10 so that the term from Theorem 7(a), $\left|\log_2|y_{q+1}| - \log_2|y_Q|\right| \leq 2$. In addition, $|\Delta y|$ fluctuates within the class $O(10^{-16})$ which is consistent with (16). Based on the figure, $\rho = 28$, which is less than 52, and this is consistent with Theorem 7.

C. OPTIMIZATION OF THE INPUT/WORKING MEMORIES AND THE LMEMD ALGORITHM

The LMEMD algorithm consists of a main procedure described in Section II and a sifting procedure called LMSIFT. The sifting in (3) can be rewritten into two steps as below.

$$\begin{aligned} \tilde{x}(t) &\leftarrow x(t) - u(t; \gamma_u, x(\gamma_u))/2; \\ \tilde{x}(t) &\leftarrow \tilde{x}(t) - l(t; \gamma_l, x(\gamma_l))/2, \end{aligned} \tag{24}$$

where γ_u and γ_l are the time coordinates of the local maxima and minima of $x(t)$ (before sifting), respectively. We first present an initial version. The sliding window w_{ui} is applied sequentially from left to right to perform the CSI of $u(t)$ by (24) as shown in Fig. 1(b). The RHS of $u(t)$ in (24) can be directly assigned to its LHS so that it is not required to store $u(t)$. Next CSI is performed for $l(t)$. Since the CSI is processed window-wise, the memory allocated in the previous window can be reused for later windows. If we choose $n_w \ll n$, the memory in the CSI is negligible.

It is required to store both $x(t)$ and $\tilde{x}(t)$ for the initial version, therefore we develop the alternative-envelope (AESW) approach that $x(t)$ and $\tilde{x}(t)$ can share the same memory. The number of local maxima for upper envelope in w_{ui} and number of local minima for lower envelope in w_{li} are chosen to be the same, n_w, but their lengths may not be identical as illustrated in Fig. 1(b). We start the analysis with $i = j = 1$. Compare γ_{ui} in w_{ui} with γ_{lj} in w_{lj}. If $\gamma_{lj} < \gamma_{ui}$, then compute $l(t)$ in w_{lj+1}; otherwise compute $u(t)$ in w_{ui+1}. The same process continues until all the windows are accessed.

Let us now explain why $x(t)$ and $\tilde{x}(t)$ can share the same memory in the AESW procedure. The $u(t)$ and $l(t)$ in (24) depend on the local extrema of $x(t)$ and the coordinates of all local extrema $\{\gamma_{lj}, \gamma_{ui}\}$ are already sorted by the AESW procedure. Then, $(\gamma_u, x(\gamma_u))$ or $(\gamma_l, x(\gamma_l))$ is computed sequentially from left to right. The results on the LHS of (24) are subsequently assigned to $x(t)$ so that it is not required to store $\tilde{x}(t)$. After (24) is completed, the value of $x(t)$ will be overwritten. The LMSIFT algorithm is listed below.

LMEMD Sifting: $\tilde{x}(t) = \text{LMSIFT}(x(t); n_R, n_w)$

1: Set $i = j = 1$.
2: Detect γ_{ui} and $x(\gamma_{ui})$ in w_{ui}, and $\gamma_{\ell j}$ and $x(\gamma_{\ell j})$ in $w_{\ell j}$
3: if $\gamma_{\ell j} < \gamma_{ui}$
  Compute $\vec{y}$ in $w_{\ell j}$ and $\tilde{x}(t) \leftarrow x(t) - l(t)/2$ in $w_{\ell cj}$.
  Set $j = j + 1$. Detect $\gamma_{\ell j}$ and $x(\gamma_{\ell j})$ in $w_{\ell j}$.
 else
  Compute $\vec{y}$ in w_{ui} and $\tilde{x}(t) \leftarrow x(t) - u(t)/2$ in w_{uci}.
  Set $i = i + 1$. Detect γ_{ui} and $x(\gamma_{ui})$ in w_{ui}.
4: Repeat Step 3 until all of windows are accessed.

The parameters n_R and n_w are chosen to be constants, and $n_w \ll n$. Next we count the memory for the LMEMD.

Part 1). Output memory. The output IMF $C_m(t)$ is stored as a float array of length n, then $M_{out} = m_{out}n$.

Part 2). CSI procedure.

(a) extrema detection: The local maxima γ_k and $x(\gamma_k)$ are stored in an n_w integer and n_w float array respectively. Assume $M_{int} = M_{float}/2$, then $M_{max} + M_{min} = 3n_w$.

(b) linear solver: Each of a_k, b_k, c_k, d_k, and y_k in (1) is stored as a float array of length n_w. Therefore, $M_{tri} = 5n_w$.

Part 3). Working memory. The residue $R(t)$ and proto-IMF $x(t)$ are stored as a float arrays of length n. Then, $M_{main} = 2n$.

The total storage is obtained by summing up all parts as follows.

$$M_{LMEMD} = (2 + m_{out})\, n + 8n_w \cong (2 + m_{out})\, n, \quad \text{given that } n_w \ll n. \tag{25}$$

Note that the input signal $x(t)$ will be overwritten when the EMD procedure is finished. If it is required to reserve the input signal, an additional float array of length n has to be invoked. The memory requirement for EMD is listed in Table 1.

Finally, we compare the memory requirements of LMEMD and that of the continuous wavelet transform (CWT). Let $\psi(t)$ be a wavelet, then the CWT of a signal $x(t)$ is written as follows:

$$g(t) = \psi(t) * x(t). \tag{26}$$

Let F denote the Fourier transform operator. To speed up the operation, (26) is often implemented using FFT as follows.

$$g(t) = F^{-1}\{F(\psi(t))F(x(t))\} \tag{27}$$

In (27), $F(\psi(t))$ and $F(x(t))$ are both arrays of complex numbers of length n so that the total memory is $4n$ (float). Compare it with (25), it is seen that the memory cost of the LMEMD is less than that of the CWT.

TABLE 1. Memory for EMD and disturbance-assisted EMDs.

Method	EMD	EEMD	CEEMD
Standard EMD	$(13 + m_{imf})n$	$(14 + m_{imf})n$	$(15 + m_{imf})n$
LMEMD	$(2 + m_{out})n$	$(3 + m_{out})n$	$(4 + m_{out})n$

D. MEMORY REQUIREMEMTS FOR EEMD AND CEEMD

We discuss memory costs for different versions of EMD algorithms in different application scenarios.

1) EEMD

Let n_e, n_s, and ε denote, respectively, the number of realizations, the number of siftings, and the disturbance amplitude. The EEMD [4] algorithm is briefly summarized below.

In the EEMD algorithm, it is required to store the input signal and the sum of the proto-IMF in each realization. Therefore, an additional two float arrays of length n are required.

EEMD: $\{C_m(t)\} = \text{EEMD}(x(t), n_m, n_s, n_e, \varepsilon)$

1: For each realization j, perturb $x(t)$ by a white noise $w_j(t)$:

$$\hat{x}_j(t) = x(t) + \varepsilon \cdot w_j(t).$$

2: Apply EMD to decompose $\hat{x}_j(t)$ into n_m IMFs, $\tilde{C}_{j,m}(t)$, with n_s siftings.

3: If $j \neq n_e$, set $j = j + 1$ and go to step 1.

4: Obtain IMFk by $C_k(t) = \sum_{j=1}^{n_e} \tilde{C}_{j,m}(t)/n_e, m = 1$ to n_m.

2) CEEMD

The CEEMD [5] is similar to the EEMD except that an assisting-noise is both added to and subtracted from the signal in each realization. Therefore, it is necessary to store the assisting noise in a float array of length n.

The memory requirements for EEMD and CEEMD are listed in Table 1.

IV. NUMERICAL EXPERIMENTS

Numerical experiments will be presented in order to confirm the claimed the memory cost and the computational time of the proposed LMEMD algorithm. The computer program is implemented in sequential C language with 64-bit float data type using a Matlab interface. Numerical experiments were conducted on a desktop computer with 3.0 GHz CPU, 16 GB DRAM memory running on Windows 10. The LMEMD computer code can be accessed at [35].

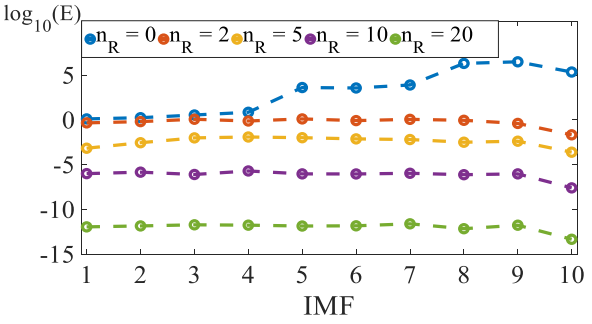

FIGURE 5. The effect of n_R on the error of the IMFs in Example 3.

Example 3: Decomposition of a white noise.

A white noise is decomposed into 10 IMFs using EMD with the criterion of 10 sifting iterations for the generation of each IMF. In practice, sometimes it is not allowed to choose $n_R \geq \rho$, which is the number that appears in Theorem 2. For example, in an online real-time computation, the number of local maxima n_R in the overlap regions actually represents the data latency. If n_R is too large, it might introduce unacceptable data latency. First, we assess the effect of number of local maxima (resp. minima), n_R, on the error of IMFm, $C_m(t)$. The maximum error E_m, for IMFm, is defined as follows

$$E_m = \left|C_m(t) - C_{ex,m}(t)\right| / \left|C_{ex,m}(t)\right|, \tag{28}$$

where $C_{ex,m}(t)$ is the exact IMFm. For this task, the IMFs obtained using offline EMD ($n_w = n$) are taken as the exact IMFs. The errors $\log_{10} E_m$ for $m = 1$ to 10 and different values of n_R are plotted in Fig. 5. It is observed that E_m decreases with n_R and $E_m 0$ for $n_R \geq \rho = 28$ (not shown in the figure), which is less than the value in Theorem 7.

In the following tests, the LMEMD is performed with $n_w = 1000$, and $n_R = 50$.

Next, we test the run time of LMEMD. Fig. 6 plots the run time t_{LMEMD} and t_{EMD} versus different data length n. It is confirmed that all the IMFs obtained by LMEMD and EMD are identical. Fig. 6 shows that t_{LMEMD} grows linearly with respect to n, which implies the computational complexity of LMEMD is the same as (the offline) EMD. The run time increment of t_{LMEMD} is within 2% because of the overhead in overlapping regions including extrema detection and tridiagonal solver.

We now turn to compare the IMFs obtained by LMEMD with those obtained by fast EMD [18] by treating the latter as the exact solutions in (28). A segment of the results is displayed in Fig. 7. It is observed that $C_m(t)$ and $C_{ex,m}(t)$ are almost identical. The maximum error E_m is around 10^{-15} and is caused by round-off error because the sequences of arithmetic operations for fast EMD in (3) and LMEMD in (24) are different.

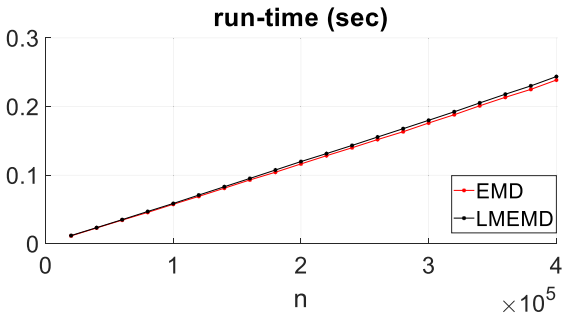

FIGURE 6. Run-time vs. data length n for the LMEMD and EMD in Example 3.

Lastly, the IMFs obtained using LMEMD are compared with those of online fast EMD with a fixed overlap length equal to 40 [18]. A segment of the results is shown in Fig. 8. The error for online EMD increases with the IMF index. For the first 3 IMFs, the results using both algorithms are essentially indistinguishable from the figure. However, the error become significant for the IMFs with index greater than 3 because there are not enough local extrema points in the overlapping region.

Example 4: Noise reduction for an ECG signal containing low frequency baseline wandering noise.

A clean ECG signal and a low frequency baseline wandering noise both sampled at 360 Hz are obtained from MIT-BIH Arrhythmia Database [36]. The noisy data is obtained by superimposing the clean data and noise as displayed in Fig. 9(a). The amplitude of the R-wave of an ECG signal

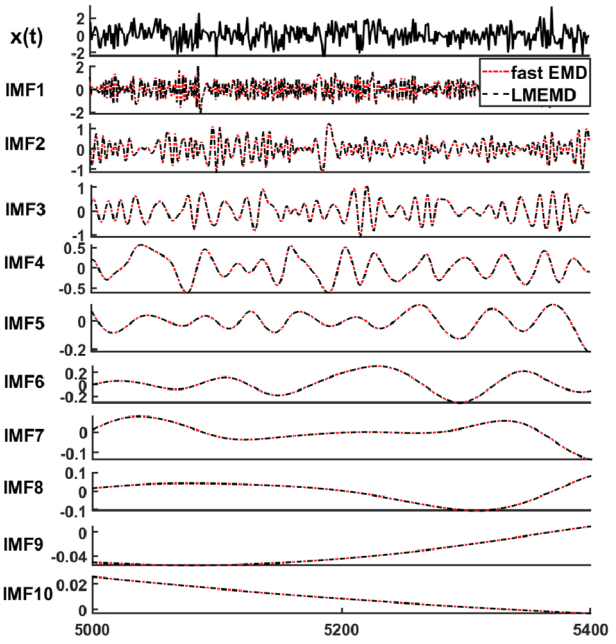

FIGURE 7. Comparison of the IMFs obtained using LMEMD and fast EMD in Example 3.

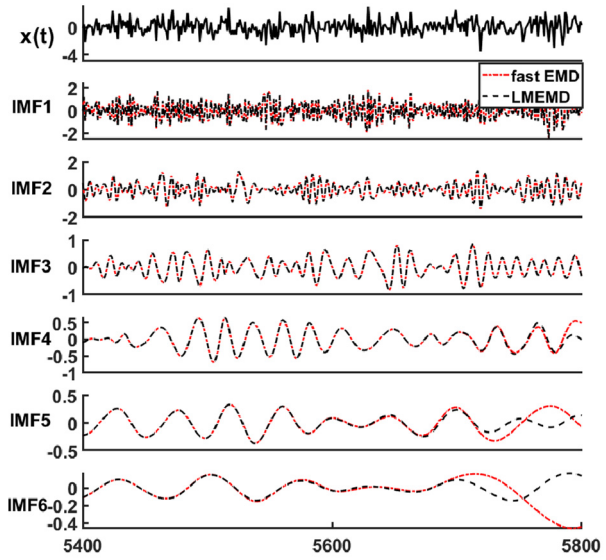

FIGURE 8. Comparison of the first 6 IMFs obtained using LMEMD with $n_R = 50$ and online fast EMD with fixed overlap length (=40) in Example 3.

(see Fig. 9(b)) is often much larger than other local maxima so that $x''(\gamma_k)$ in (21) and y_k in (9) at the peaks of the R waves are much greater than those of other local maxima. We investigate how the term $\log_2 |y_{q+1}/y_Q|$ will affect the value of ρ in Theorem 7.

The signal is decomposed using EEMD with $m_{imf} = 8$, sifting number $n_s = 10$, realization number $n_e = 400$, and noise level $\varepsilon = 0.3$. The LMEEMD is applied with $n_C = 500$, and $n_R = 50$.

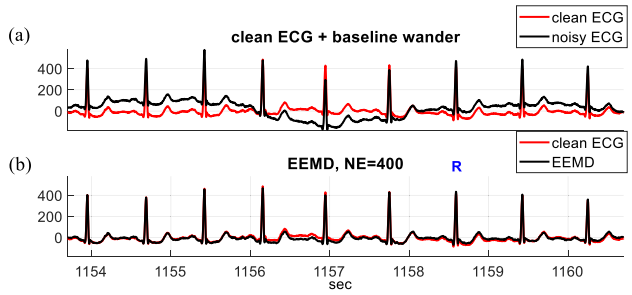

FIGURE 9. Profile of the ECG signal in Example 4. (a) noisy and clean signal; (b) The denoised signal using LMEEMD.

The "clean" ECG mode is obtained as the summation of IMF2 to IMF7, i.e., $PO_1 = \sum_{m=2}^{7} C_m(t)$, so that $m_{out} = 1$. The denoised ECG is shown in Fig. 9(b). It can be seen that the low frequency noise is almost completely removed. It is found that as $n_R \geq \rho = 28$ (not shown in the figure), the results obtained using EEMD and LMEEMD are identical and the value of ρ is less than the upper bound of Theorem 7(b).

The memory cost for LMEEMD and EEMD are $4n$ and $22n$, respectively, which indicates the memory saving is over 80%.

We then test the run time of LMEEMD. Fig. 10 displays the run time t_{LMEEMD} and t_{EEMD} versus different data length n. It has been confirmed numerically that all the IMFs obtained by LMEMD and EMD are identical. The run time increment of t_{LMEEMD} is under 3%.

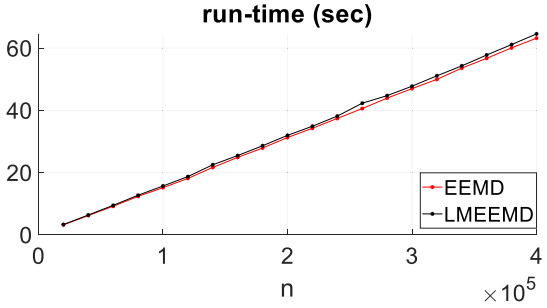

FIGURE 10. Run-time vs. data length n for the LMEEMD and EEMD in Example 4.

V. CONCLUSION

In this study, we prove that the memory cost of EMD can be improved from $(13 + m_{imf})\, n$ to $(2 + m_{out})\, n$ by adopting LMEMD (where n is the signal length, m_{imf} is the number of IMFs, and $m_{out} \leq m_{imf}$, where the former is the number of target physical oscillation modes), while giving the same results and efficacy. It indicates that EMD is in essence not only a computationally efficient method but is also a low-memory algorithm. Our findings also imply that the memory cost of the EMD is less than that of the wavelet transform implemented by fast Fourier transform. The method of LMEMD can be also applied to other EMD-based algorithms. This new result is a solid evidence of that the EMD-based algorithm can be applied in a wider range of fields of applications than so far perceived. In order to perform some real-time computation in applications, we will implement the LMEMD/LMEEMD in a computation and memory limited micro-controller in the near future.

REFERENCES

[1] 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, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 454, pp. 903–995, Mar. 1971.

[2] G. Rilling and P. Flandrin, "One or two frequencies? The empirical mode decomposition answers," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 85–95, Jan. 2008.

[3] T. Oberlin, S. Meignen, and V. Perrier, "An alternative formulation for the empirical mode decomposition," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2236–2246, May 2012.

[4] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, 2009.

[5] J.-R. Yeh, J.-S. Shieh, and N. E. Huang, "Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method," *Adv. Adapt. Data Anal.*, vol. 2, no. 2, pp. 135–156, Apr. 2010.

[6] R. Deering and J. F. Kaiser, "The use of a masking signal to improve empirical mode decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 4, Mar. 2005, pp. iv/485–iv/488.

[7] Y.-H. Wang, K. Hu, and M.-T. Lo, "Uniform phase empirical mode decomposition: An optimal hybridization of masking signal and ensemble approaches," *IEEE Access*, vol. 6, pp. 34819–34833, 2018.

[8] N. Rehman and D. P. Mandic, "Filter bank property of multivariate empirical mode decomposition," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2421–2426, May 2011.

[9] Y. Lv, R. Yuan, and G. Song, "Multivariate empirical mode decomposition and its application to fault diagnosis of rolling bearing," *Mech. Syst. Signal Process.*, vol. 81, pp. 219–234, Dec. 2016.

[10] T.-T. Tran, V.-T. Pham, C. Lin, H.-W. Yang, Y.-H. Wang, K.-K. Shyu, W.-Y.-I. Tseng, M.-Y.-M. Su, L.-Y. Lin, and M.-T. Lo, "Empirical mode decomposition and monogenic signal-based approach for quantification of myocardial infarction from MR images," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 2, pp. 731–743, Mar. 2019.

[11] K. Hu, C. K. Peng, N. E. Huang, Z. Wu, L. A. Lipsitz, J. Cavallerano, and V. Novak, "Altered phase interactions between spontaneous blood pressure and flow fluctuations in type 2 diabetes mellitus: Nonlinear assessment of cerebral autoregulation," *Phys. A, Stat. Mech. Appl.*, vol. 387, no. 10, pp. 2279–2292, Apr. 2008.

[12] H.-W. Yang, S.-K. Jeng, H.-W. V. Young, C. Lin, Y.-H. Wang, K. Hu, and M.-T. Lo, "A minimum arclength method for removing spikes in empirical mode decomposition," *IEEE Access*, vol. 7, pp. 13284–13294, 2019.

[13] R. Yuan, Y. Lv, H. Li, and G. Song, "Robust fault diagnosis of rolling bearings using multivariate intrinsic multiscale entropy analysis and neural network under varying operating conditions," *IEEE Access*, vol. 7, pp. 130804–130819, 2019.

[14] Y. Li, M. Xu, X. Liang, and W. Huang, "Application of bandwidth EMD and adaptive multiscale morphology analysis for incipient fault diagnosis of rolling bearings," *IEEE Trans. Ind. Electron.*, vol. 64, no. 8, pp. 6506–6517, Aug. 2017.

[15] X. Yan, Y. Liu, W. Zhang, M. Jia, and X. Wang, "Research on a novel improved adaptive variational mode decomposition method in rotor fault diagnosis," *Appl. Sci.*, vol. 10, no. 5, p. 1696, Mar. 2020.

[16] P. Pandey, N. D. Bokde, S. Dongre, and R. Gupta, "Hybrid models for water demand forecasting," *J. Water Resour. Planning Manage.*, vol. 147, no. 2, Feb. 2021, Art. no. 04020106.

[17] P. J. J. Luukko, J. Helske, and E. Räsänen, "Introducing libeemd: A program package for performing the ensemble empirical mode decomposition," *Comput. Stat.*, vol. 31, no. 2, pp. 545–557, 2016.

[18] Y.-H. Wang, C.-H. Yeh, H.-W. V. Young, K. Hu, and M.-T. Lo, "On the computational complexity of the empirical mode decomposition algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 400, pp. 159–167, Apr. 2014.

[19] Y. Chen, "Dip-separated structural filtering using seislet transform and adaptive empirical mode decomposition based dip filter," *Geophys. J. Int.*, vol. 206, no. 1, pp. 457–469, Apr. 2016.
[20] P.-Y. Chen, Y.-C. Lai, and J.-Y. Zheng, "Hardware design and implementation for empirical mode decomposition," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3686–3694, Jun. 2016.
[21] Z. Liu, Y. Kong, X. Zhang, J. Wu, and W. Lu, "Vital sign extraction in the presence of radar mutual interference," *IEEE Signal Process. Lett.*, vol. 27, pp. 1745–1749, 2020.
[22] Y.-H. Wang, I.-Y. Chen, H. Chiueh, and S.-F. Liang, "A low-cost implementation of sample entropy in wearable embedded systems: An example of online analysis for sleep EEG," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
[23] S. Asano, T. Maruyama, and Y. Yamaguchi, "Performance comparison of FPGA, GPU and CPU in image processing," in *Proc. Int. Conf. Field Program. Log. Appl.*, Aug. 2009, pp. 126–131.
[24] T. Mujahid, A. U. Rahman, and M. M. Khan, "GPU-accelerated multivariate empirical mode decomposition for massive neural data processing," *IEEE Access*, vol. 5, pp. 8691–8701, 2017.
[25] D. Chen, D. Li, M. Xiong, H. Bao, and X. Li, "GPGPU-aided ensemble empirical-mode decomposition for EEG analysis during anesthesia," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 6, pp. 1417–1427, Nov. 2010.
[26] G. Rilling, P. Flandrin, and P. Goncalves, "On empirical mode decomposition and its algorithms," in *Proc. IEEE-EURASIP Workshop Nonlinear Signal Image Process. (IEEER Grado)*, Jun. 2003, pp. 8–11.
[27] R. Faltermeier, A. Zeiler, I. R. Keck, A. M. Tome, A. Braw, and E. W. Lang, "Sliding empirical mode decomposition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.
[28] W.-C. Shen, H.-I. Jen, and A.-Y. Wu, "New Ping-Pong scheduling for low-latency EMD engine design in Hilbert–Huang transform," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 8, pp. 532–536, Aug. 2013.
[29] R. Fontugne, P. Borgnat, and P. Flandrin, "Online empirical mode decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 4306–4310.
[30] A. Stallone, A. Cicone, and M. Materassi, "New insights and best practices for the successful use of empirical mode decomposition, iterative filtering and derived algorithms," *Sci. Rep.*, vol. 10, no. 1, pp. 1–15, Dec. 2020.
[31] J. Gilles, "Empirical wavelet transform," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 3999–4010, Aug. 2013.
[32] Y.-H. Wang and S.-H. Cheng, "Boundary effects for EMD-based algorithms," *IEEE Signal Process. Lett.*, vol. 29, pp. 1032–1036, 2022.
[33] Y.-H. Wang, H.-W. V. Young, and M.-T. Lo, "The inner structure of empirical mode decomposition," *Phys. A, Stat. Mech. Appl.*, vol. 462, no. 300, pp. 1003–1017, 2016.
[34] M. A. Motin, C. K. Karmakar, and M. Palaniswami, "Selection of empirical mode decomposition techniques for extracting breathing rate from PPG," *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 592–596, Apr. 2019.
[35] [Online]. Available: https://sites.google.com/view/yhw-personal-homepage/
[36] A. L. Goldberger, L. A. N. Amaral, L. Glass, and J. M. Hausdorff, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

HSU-WEN VINCENT YOUNG received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the Ph.D. degree in mathematics from the University of Michigan, Ann Arbor, MI, USA.

He completed Postdoctoral training in mathematics at National Taiwan University and in adaptive and nonlinear data analysis at National Central University, Taiwan. He is currently an Assistant Professor of Electronic Engineering with Chung Yuan Christian University, Taoyuan, Taiwan. His current research interests include data analysis, theoretical problems in data science and signal processing, mathematical modeling, and algebraic geometry and commutative algebra with an eye towards applications.

YU-CHUAN LIN received the B.S. degree in materials science and engineering from National United University (NUU), Miaoli, Taiwan, in 2020, and the M.S. degree in mechanical engineering from National Central University (NCU), Taoyuan, Taiwan, in 2022. His research interests include signal processing, firmware, and computer science.

YUNG-HUNG WANG received the B.S. degree in naval architecture from the National Taiwan University, Taipei, Taiwan, the M.S. degree in mechanical engineering from UC Santa Barbara, CA, USA, and the Ph.D. degree in aeronautics and astronautics from Stanford University, Stanford, CA, USA.

He is currently an Associate Professor with the Department of Mechanical Engineering, National Central University, Taiwan. His research interests include signal processing, fluid mechanics, biomedical engineering, and electronic design automation.