

Uyarlanırlı Alçak Geçiren Süzme Yardımlı Ampirik Mod Ayırma Empirical Mode Decomposition Aided by Adaptive Low Pass Filtering

Onur Ozturk, Orhan Arıkan, A. Enis Cetin

Elektrik ve Elektronik Mühendisliđi Bölümü
Bilkent Üniversitesi

ozturk,oarikan@ee.bilkent.edu.tr, cetin@bilkent.edu.tr

ÖZETÇE

Ampirik Mod Ayırma (AMA), taban fonksiyonlarını iřaretin kendisinden türeten uyarlanırlı bir iřaret çözümlenme tekniđidir. AMA, yerel ortalama hesabını gerektiren bir dizi ayıklama iřlemi döngüsü sayesinde gerçekenir. Bu amaçla, iřaretin yerel minimum ve maksimum noktalarının uygun zaman ölçeklerini oluşturduđu kabul edilir. AMA, yapay uçdeđerler yaratan gürültü altında mod karışması adı verilen bir olguya maruz kalarak doğruluđunu kaybeder. Bu makalede, mod karışmasını engellemek için her yerel ortalama hesaplama adımından önce iřareti Ayırık Kosinüs Dönüşümü bölgesinde uyarlanırlı bir şekilde süzmeyi öneriyoruz. Gürültü ayıklama süzgecinin eřiđi, süzme sonrasında korunan iřaret enerjisinin ve atılan uç nokta sayısının bir fonksiyonu olan çarpma formunda bir ölçüt için eniyilenmektedir. Sentetik iřaretlerden elde edilen sonuçlar önerilen tekniđin potansiyelini göstermektedir.

ABSTRACT

Empirical Mode Decomposition (EMD) is an adaptive signal analysis technique which derives its basis functions from the signal itself. EMD is realized through successive iterations of a sifting process requiring local mean computation. For that purpose, local minima and maxima of the signal are assumed to constitute proper local time scales. EMD lacks accuracy, however, experiencing the so-called mode mixing phenomenon in the presence of noise which creates artificial extrema. In this paper, we propose adaptively filtering the signal in Discrete Cosine Transform domain before each local mean computation step to prevent mode mixing. Denoising filter thresholds are optimized for a product form criterion which is a function of the preserved energy and the eliminated number of extrema of the signal after filtering. Results obtained from synthetic signals reveal the potential of the proposed technique.

1. GİRİŐ

İřaret iřleme literatüründe, teorik temelleri iyi anlařıldıđı, gürültü altında gürbüz sonuçlar verdiđi ve verimli gerçekenleřtirmeleri bulunduđu için Fourier ve dalgacık dönüşümleri sıklıkla kullanılmaktadır. Ancak, bu iki dönüşüm tekniđinde de iřaret kendisinden bađımsız olarak belirlenmiř

taban fonksiyonları cinsinden ifade edilir. Ampirik Mod Ayırma (AMA) tekniđinde ise, [1], durađan ve dođrusal olmayan süreçlerin ürettiđi iřaretlerin uyarlanırlı olarak bir dizi genlik ve frekans kiplemeli içsel mod fonksiyonuna (*IMF*) ayrılması sađlanır. *IMF* ařađıda verilen iki kořulu sađlayan herhangi bir fonksiyon olarak tanımlanmaktadır:

K1: Sıfır kesiř sayısı ile yerel uç nokta sayısı arasındaki fark en fazla 1'dir[†].

K2: Yerel ortalaması sıfırdır.

K1 ve K2 kořullarının sađlanması, *IMF*'ler üzerinden anlık frekans hesaplanabilmesi açısından gereklidir [1]. K2'de geçen yerel ortalamasının hesaplanması için yerel bir zaman ölçeđinin belirlenmesi gerekmektedir. AMA algoritmasında, iřaretin yerel minimum ve maksimum noktaları bu amaçla kullanılmaktadır. Bu çalışmada da gösterildiđi gibi makul sayılabilecek gürültü seviyelerinde bile gürültünün neden olduđu yapay uç noktalar nedeniyle üretilen *IMF* kümesi *mod karışması* adı verilen bir olguyu göstermektedir. Mod karışması farklı ölçeklerde iřaret bileřenlerinin aynı, benzer ölçeklerde iřaret bileřenlerinin ise farklı *IMF*'lerde toplanması olarak ifade edilebilir. Mod karışmasına uğramıř *IMF*'ler, söz konusu *IMF* kořullarını sađlayabilseler de fiziksel olarak anlamlı anlık frekans bilgisi elde edilmesini engellerler.

Mod karışmasıyla mücadele konusunda çeřitli çalışmalar yapılmıřtır, [2]-[4]. Bunların içinde öne çıkanlardan biri olan [4]'te gürültü yardımıyla AMA algoritmasını iyileřtirmeye çalışılan Toplu Ampirik Mod Ayırma (TAMA) tekniđi önerilmiřtir. Bu çalışmada ise AMA algoritmasındaki yerel ortalama hesaplama adımından önce uyarlanırlı alçak geçiren süzme önerilmektedir.

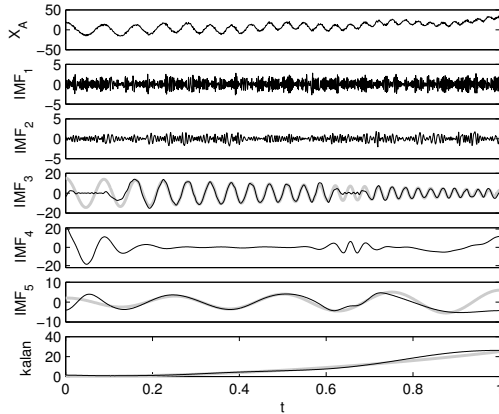
Bölüm 2 ve Bölüm 3'te sırasıyla AMA ve TAMA algoritmaları; Bölüm 4'te ise AMA algoritmasına yönelik önerilen iyileřtirme anlatılmaktadır. AMA, çeřitli tipte iřaretlere uygulanabilir bir teknik olmasına rađmen, söz konusu algoritmalar bu bölümlerde iřaret iřleme problemlerinde sıklıkla karřılařılan sinüs ve cıvıltı formundaki iřaretlerle denenmiřtir. Son olarak, Bölüm 5'te bu denemelerden elde edilen sonuçlar deđerlendirilmektedir.

[†]Bu kořul, 0 noktası üzerinde herhangi bir yerel minimum noktasının; sıfır noktası altında ise herhangi bir yerel maksimum noktasının bulunmamasını sađlar.

Bu çalışma AB tarafından FIRESENSE (FP7-ENV-2009-1) projesi kapsamında desteklenmektedir.

2. Ampirik Mod Ayırma (AMA) Algoritması

Algoritma 1'de adımları detaylı bir şekilde gösterilen AMA, her defasında yerel ortalamasının işaretin kendisinden çıkarıldığı ayıklama döngüleri sayesinde çok bileşenli bir işareti tek bileşenli IMF 'lere ayırmaya çalışır. 16-20 arasındaki adımlarda gösterildiği gibi işaretin yerel maksimum ve minimum noktaları ve kübik eğri ara değerlendirme yöntemi kullanılarak sırasıyla $ustZarf$ ve $altZarf$ adı verilen iki işaret elde edilir. Bu iki işaretin noktasal ortalaması da yerel ortalama kabul edilir. Hem kullanılan ara değerlendirme yönteminin neden olabileceği üste ve alta sapmalar, hem de yapılan yerel ortalama tanımının ideal olmaması nedeniyle, işaret için tekrar hesaplanacak yerel ortalamasının sıfır olmadığı durumlarla karşılaşılabilir. Bu nedenle ayıklama işlemi, adım 6'da kontrolü yapılan bir başarı ölçütü sağlanana kadar devam ettirilir (Bu çalışmada, [4]'te önerilen sabit (=10) döngü sayısı kullanılmıştır.). Ölçüt sağlandığında ise elde edilen işaret geçerli bir IMF kabul edilip orijinal işaretten çıkarılır. Ayıklama işlemi, çıkarmadan sonra kalan işaret üzerinden tekdüze artan ya da azalan bir işaret elde edene kadar devam ettirilir. AMA sonrasında elde edilen IMF 'ler ayıklama işleminin doğası gereği düşük ölçekten yüksek ölçğe doğru sıralanır.



Şekil 1: X_A İşaretinden (20dB SNR_g) AMA Algoritmasının Ürettiği IMF 'ler ve $kalan$ İşaret (8.75dB SNR_c). (X_A 'nın Tablo 1'deki bileşenleri gri ile gösterilmiştir.)

Tablo 1: Sırasıyla X_A ve X_B İşaretlerinin Bileşenlerini Oluşturan MATLAB Komutları ($t = 0 : 0.001 : 1$)

<pre>chirp(t,10,1,40).*linspace(15,3,length(t)) cos(2*pi*4.*t).*linspace(2,6,length(t)) linspace(0,5,length(t)).^2</pre>
<pre>chirp(t,30,1,80).*sin(2*pi*2.*t) chirp(t,5,1,20).*cos(2*pi*2.*t) linspace(-1,1,length(t)).^3</pre>

Algoritma 1 AMA Algoritması

```

1: fonksiyon: AMA( $x$ )
2:    $i \leftarrow 0$ 
3:    $kalan \leftarrow x$ 
4:   while HAYIR ==  $tekduze\_mi(kalan)$  do
5:      $IMF \leftarrow kalan$ 
6:     while HAYIR ==  $IMF\_mi(IMF)$  do
7:        $yerelOrtalama \leftarrow YERELORT(IMF)$ 
8:        $IMF \leftarrow IMF - yerelOrtalama$ 
9:     end while
10:     $kalan \leftarrow kalan - IMF$ 
11:     $y(i) \leftarrow IMF$ 
12:     $i \leftarrow i + 1$ 
13:  end while
14: end fonksiyon:

15: fonksiyon: YERELORT( $IMF$ )
16:    $maksNoktalar \leftarrow maksimumBul(IMF)$ 
17:    $minNoktalar \leftarrow minimumBul(IMF)$ 
18:    $ustZarf \leftarrow kubikEgri(maksNoktalar)$ 
19:    $altZarf \leftarrow kubikEgri(minNoktalar)$ 
20:    $yerelOrtalama \leftarrow (ustZarf + altZarf)/2$ 
21: end fonksiyon:

```

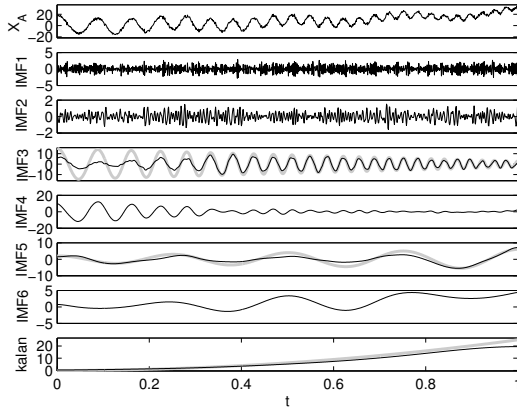
3. Toplu Ampirik Mod Ayırma (TAMA) Algoritması

AMA algoritmasıyla çözümlenecek işaretin üzerine gürültü bindirildiğinde ortaya çıkan yapay uç noktalar nedeniyle yerel ortalama hesapları bozulmakta ve üretilen IMF 'ler Şekil 1'deki gibi mod karışmasına uğramaktadır. Bu örnekte, bileşenleri Tablo 1'de verilen X_A işaretinden üretilen IMF_3 ve IMF_4 işaretleri istenmeyen bir şekilde benzer ölçeklere sahiptir. Mod karışmasıyla mücadele etmek için [4]'te; AMA algoritmasını bağımsız özdeşçe dağılmış beyaz gürültü eklenmiş girdilerle birçok kere çalıştırmak ve ortaya çıkan IMF 'lerin ortalamasını almak önerilmiştir. Eklenen beyaz gürültünün [5]'teki bir gözlemden yola çıkarak AMA algoritmasına çözümlediği işareti ölçeklere ayırmada referans sağlaması ve bu sayede mod karışmasını önlemesi amaçlanmıştır. Toplu AMA (Algoritma 2) adı verilen bu algoritmanın Tablo 1'de verilen X_A ve X_B işaretlerinden ürettiği IMF 'ler sırasıyla Şekil 2 ve Şekil 3'te verilmiştir. Üretilen IMF 'lerde mod karışmasının engellenemediği sırasıyla $IMF_3 - IMF_4$ ve $IMF_2 - IMF_3 - IMF_4$ işaretlerinin ölçekleri kendi aralarında karşılaştırıldığında görülebilmektedir.

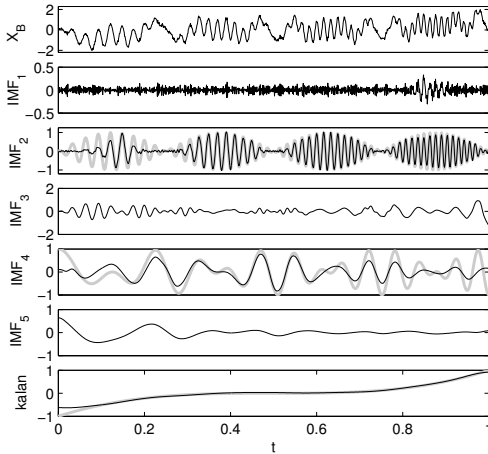
Bu çalışmada mod ayırma başarımı, giriş SNR değeri SNR_g 'ye karşılık gelen ve (1)'de tanımlanan çıkış SNR değeri SNR_c ile ölçülmektedir. Bu denklemde; $P[\cdot]$ argümanının gücünü bulan operatörü, X gürültüsüz giriş işaretini, E_i ise X 'in bileşenleri ve karşılık gelen IMF ve $kalan$ işaretleri arasındaki fark işaretlerini temsil etmektedir.

$$SNR_c := 10 \log_{10} \left(\frac{P[X]}{\sum_i P[E_i]} \right) \quad (1)$$

4. Uyarlanırlı Alçak Geçiren Süzme Yardımlı Ampirik Mod Ayırma (AMA) Algoritması



Şekil 2: X_A İşaretinden (20dB SNR_g) TAMA Algoritmasının ($Gstd = 0.1, T = 1000$) Ürettiği IMF 'ler ve $kalan$ İşaret (9.86dB SNR_c). (X_A 'nın Tablo 1'deki bileşenleri gri ile gösterilmiştir.)



Şekil 3: X_B İşaretinden (20dB SNR_g) TAMA Algoritmasının ($Gstd = 0.1, T = 1000$) Ürettiği IMF 'ler ve $kalan$ İşaret (6.87dB SNR_c). (X_B 'nin Tablo 1'deki bileşenleri gri ile gösterilmiştir.)

Algoritma 2 TAMA Algoritması

```

1: fonksiyon: TAMA( $x, Gstd, T$ )
2:    $y \leftarrow 0$ 
3:    $norm_x \leftarrow norm(x)$ 
4:    $x \leftarrow x/norm_x$ 
5:   for  $i \leftarrow 0, T - 1$  do
6:      $x \leftarrow x + Gstd * beyazGurultu$ 
7:      $y \leftarrow y + AMA(x)$ 
8:   end for
9:    $y \leftarrow y * norm_x/T$ 
10: end fonksiyon:

```

Bu bölümde Algoritma 3 adım 7'de yapılması önerilen süzme önışlemi anlatılmaktadır. IMF işareti doğrudan yerel ortalama hesaplamasında kullanıldığı zaman gürültünün ortaya çıkardığı yapay uç noktalar mod karışmasına yol açmaktadır. Bu gözlemden yola çıkarak Fourier, dalgacık ya da herhangi bir dönüşüm bölgesinde IMF işaretinin yerel ortalama hesabından önce gürültü ayıklamaya tabi tutulması önerilmektedir. Bu çalışmada, gerçel sayılı işaretler düşünülerek Ayrık Kosinüs Dönüşümü (AKD) tercih edilmiştir.

IMF , IMF 'nin AKD bölgesindeki karşılığı olsun. N , IMF 'nin uzunluğu ise; dönüşüm uzunluğu K , AKD'nin verimli bir şekilde gerçekleştirilebilmesi için $2^{\lceil \log_2 N \rceil}$ alınabilir. AKD bölgesinde süzme işleminde kullanılan k ($k \in [0, K]$) uzunluğundaki pencereye W_k ; süzme işleminden sonra elde edilen işarete de $IMF_k = IMF * W_k$ diyelim. $E[\cdot]$ ve $U[\cdot]$ sırasıyla argümanlarının enerjisini ve yerel minimum ve yerel maksimum nokta sayılarının minimumunu veren operatörler olsun. AKD bölgesindeki süzme işleminde kullanılan pencere uzunluğu, k , (2)'de verilen çarpma formundaki ölçüt için eniyilebilir.

$$\underset{k}{\operatorname{argmax}} \left(\frac{E[IMF_k]}{E[IMF]} * \left(1 - \frac{U[IMF_k]}{U[IMF]} \right) \right) \quad (2)$$

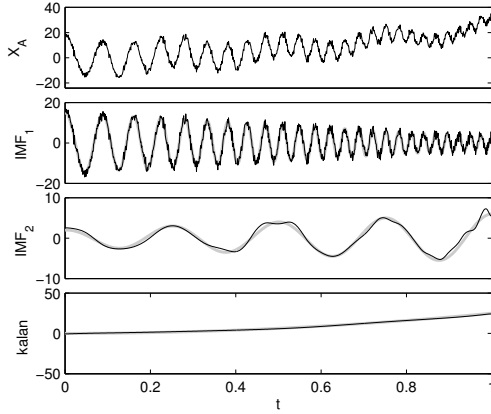
(2)'de kullanılan IMF_k, IMF_k 'nin Ters AKD'si alınarak bulunan işarettir. Bu ölçüt, süzme işlemi sonrasında IMF 'in uç nokta sayısını, dolayısıyla yapay uç noktasını azaltmayı; bunu yaparken de enerjisini, dolayısıyla işaretin ölçeklerini korumayı amaçlamaktadır. Ölçütün değeri $k = 0$ ve $k = K$ için 0; $k \in (0, K)$ için ise 0 ve 1 arasında olduğundan, her IMF için mutlak maksimum değerini verecek bir k değeri bulunabilir. Şekil 4 ve Şekil 5'te bu ölçüt kullanılarak sırasıyla X_A ve X_B işaretlerinden elde edilen IMF 'ler gösterilmiştir. Bu örneklerde, mod karışması önerilen yöntemle engellenebilmiştir.

Algoritma 3 Uyarlanırlı Alçak Geçiren Süzme Yardımlı AMA Algoritması

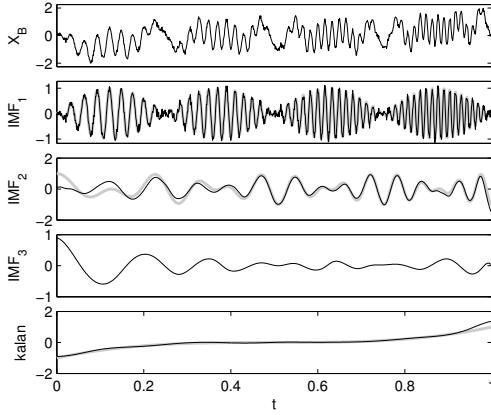
```

1: fonksiyon: AMA( $x$ )
2:    $i \leftarrow 0$ 
3:    $kalan \leftarrow x$ 
4:   while HAYIR ==  $tekduze_mi(kalan)$  do
5:      $IMF \leftarrow kalan$ 
6:     while HAYIR ==  $IMF_mi(IMF)$  do
7:        $IMF_{AG} \leftarrow SUZME(IMF)$ 
8:        $yerelOrtalama \leftarrow YERELORT(IMF_{AG})$ 
9:        $IMF \leftarrow IMF - yerelOrtalama$ 
10:    end while
11:     $kalan \leftarrow kalan - IMF$ 
12:     $y(i) \leftarrow IMF$ 
13:     $i \leftarrow i + 1$ 
14:  end while
15: end fonksiyon:

```



Şekil 4: X_A İşaretinden (20dB SNR_g) (2)'de Verilen Ölçüt Eniyiliyenerek Üretilen IMF 'ler ve $kalan$ İşaret (18.53dB SNR_c). (X_A 'nın Tablo 1'deki bileşenleri gri ile gösterilmiştir.)



Şekil 5: X_B İşaretinden (20dB SNR_g) (2)'de Verilen Ölçüt Eniyiliyenerek Üretilen IMF 'ler ve $kalan$ İşaret (9.69dB SNR_c). (X_B 'nin Tablo 1'deki bileşenleri gri ile gösterilmiştir.)

5. Sonuç

TAMA ve Uyarlanı Alçak Geçiren Süzme Yardımlı AMA algoritmalarının X_B işaretinin değişen SNR_g değerlerindeki başarımları Tablo 2'de verilmektedir. Bu örnek için, önerilen tekniğin yüksek SNR_g değerlerinde TAMA algoritmasından daha iyi sonuç verdiği görülmektedir. Öte yandan TAMA algoritması, düşük SNR_g değerlerinde bileşenlerden gürültüyü ayırarak giriş SNR 'larından daha yüksek çıkış SNR 'larına ulaşabilmektedir.

Süzme tekniğinin doğal bir sonucu olarak, Şekil 4 ve Şekil 5'te görüldüğü gibi, gürültü işareti üretilen IMF 'lerden ayırlanamakta, özellikle ilk IMF içinde hapsolmaktadır. Yine aynı şekillerden yola çıkarak, üretilen IMF 'lerin K1 ve K2 koşullarını sağlayamayabilecekleri de söylenebilir. Bu iki problem de, üretilen IMF 'lerin uygun bir artışlemeye tabi tu-

Tablo 2: X_B 'nin 0dB-25dB Arasındaki SNR_g (dB) Değerleri İçin Sırasıyla TAMA ($Gstd = 0.1$, $T = 1000$) ve Uyarlanı Alçak Geçiren Süzme Yardımlı AMA Algoritmalarının SNR_c (dB) Değerleri

	0	5	10	15	20	25
	3.78	4.77	5.04	7.05	6.87	6.36
	0.50	1.55	6.92	9.13	9.69	10.48

lması gerektiğini göstermektedir. Süzme işleminin performansını arttırmak amacıyla Ayrık Kosinüs Dönüşümü yerine dalgacık dönüşümü kullanılması araştırılması gereken bir konu olarak görülmektedir. İşaret ve gürültü istatistiklerinin bilindiği durularda, gürültü ayıklamak için Wiener süzgecinden de faydalanılabilir, ancak AMA algoritmasının ayıklama döngüleri sırasında bu işaretlerin istatistiklerinin değişebileceği dikkate alınmalıdır.

Bu makalede, gürültünün yol açtığı mod karışmasını engellemek için, AMA algoritmasındaki yerel ortalama hesabından önce uyarlanı alçak geçiren süzgeç kullanılması önerilmiş ve TAMA algoritmasıyla karşılaştırmalar yapılmıştır. Denenen sınırlı sinüs ve cıvıltı formundaki işaretler, Uyarlanı Alçak Geçiren Süzme Yardımlı AMA algoritmasının mod karışmasına karşı mücadelede etkin olabileceğini göstermektedir. AMA, analitik temelleri henüz kurulamamış bir algoritma olduğu için, bu konuda yapılan çalışmaların kapsamlı örneklerle desteklenmesi gerekmektedir. Önerilen tekniğin performans sınırlarının çizilebilmesi için de; daha fazla sentetik işaret ve pratik uygulama üzerinde, çeşitli SNR değerlerinde ve giriş işaret gürültüsünün beyaz olduğu durumların yanı sıra renkli olduğu durumlarda da test edilmesi gerekmektedir.

6. KAYNAKÇA

- [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," Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1971, pp. 903-995, Mar. 1998.
- [2] Gao, Yunchao; Ge, Guangtao; Sheng, Zhengyan; Sang, Enfang; , "Analysis and Solution to the Mode Mixing Phenomenon in EMD," Image and Signal Processing, 2008. CISP '08. Congress on , vol.5, no., pp.223-227, 27-30 May 2008
- [3] Huang, Norden E.; Shen, Zheng; Long, Steven R., "A New View of Nonlinear Water Waves: The Hilbert Spectrum," Annual Review of Fluid Mechanics, vol. 31, pp. 417-457
- [4] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a Noise-Assisted data analysis method," Advances in Adaptive Data Analysis, vol. 1, no. 1, pp. 1-41, 2009
- [5] P. Flandrin, G. Rilling and P. Goncalves, "Empirical mode decomposition as a filter bank," IEEE Signal Processing Letters, pp. 112-114, 2004