Using Apodization to improve the performance of the Complex Spectral Phase Estimation (CSPE) Algorithm

Jian Wang[#1], Joseph Timoney[#2], Matthieu Hodgkinson[#3]

[#]*Computer Science Department,*
National University of Ireland, Maynooth, Co. Kildare, Ireland
[1]jwang@cs.nuim.ie
[2]jtimoney@cs.nuim.ie
[3]matthew.hodgkinson@nuim.ie

***Abstract*— The recently introduced Complex Spectral Phase Evolution (CSPE) algorithm is a super-resolution technique for the estimation of the exact frequency values of sinusoidal components in a signal. However, if a component of the signal does not exist within the entire data set, it cannot be identified out by CSPE algorithm, even though it still may be visible in the FFT magnitude spectrum. In this paper, we identify the source of this problem and propose a novel approach to resolve this issue. Specifically, we will show how to use a window apodization function to improve the CSPE algorithm. Experimental results are presented to illustrate the performance enhancement.**

***Keywords*— CSPE, Apodization, Kaiser Window**

I. INTRODUCTION

Most often the estimation of the frequencies of a signal composed of sinusoidal components is done in the frequency domain using peak-picking from the magnitude spectrum of the signal. However, this accuracy of this approach is severely limited to cases where a component frequency is not a multiple of the windowed signal length divided by the sampling frequency. In essence, this means only when a component frequency is aligned exactly with the analysis frequencies of the DFT, can it be measured accurately. When the component frequency does not satisfy this constraint, a common solution that is used in Sinusoidal Modelling algorithms is to apply quadratic interpolation to the component spectral magnitudes immediately either side of the true frequency to find the correct frequency and magnitude values. However, the performance of this method is highly dependent on the window function used [1] and the length of the data for analysis. The CSPE algorithm was introduced by [2] as a method to accurately estimate the frequency of components that exist within a short time frame. It was also designed to be computationally efficient. It is actually related in some aspects to the cross-spectrogram technique of [3].

However, the CSPE algorithm has been found to be unable to detect frequency components that do not appear throughout the entire signal source under analysis. This is puzzling because an associated peak can still appear for the component in the FFT magnitude spectrum. To resolve this issue it is necessary to investigate the CSPE algorithm in more detail and determine how it can be improved.

This paper is organized as follows: Firstly, we give a general introduction to the CSPE algorithm followed by an experimental evaluation that compares the CSPE algorithm with the widely-used frequency estimation method introduced in [4]. Then, we will explain in more detail the problem of identifying components that do not exist for the complete data frame and introduce the idea of apodization to solve it. Lastly, we will show the improvement to the CSPE result by using the apodization function by providing some experimental results.

II. CSPE AND ITS COMPARISON WITH ANOTHER FREQUENCY ESTIMATION APPROACH

The principal of CSPE algorithm can be described as below:

An FFT analysis is performed twice; firstly on the signal of interest, and the second time upon the same signal but shifted in time by one sample. Then, by multiplying the sample-shifted FFT spectrum with the complex conjugate of the initial FFT spectrum, a frequency dependent function is formed from which the exact values of the frequency components it contains can be detected. This frequency dependent function has a staircase-like appearance where the flat parts of the graph indicate where the exact frequencies of the components. The width of the flat parts is dependent on the main-lobe width of window function used to select the signal before FFT processing. Mathematically, the algorithm can be described as follows:

Assuming a real signal s_0, and a one-sample shifted version of this signal s_1. Say that its frequency is $\beta = q + \delta$ where q is an integer and δ is a fractional number. If b is an initial phase, w_n is the window function used in the FFT, F_{ws_0} is windowed Fourier transform of s_0, and F_{ws_1} is the windowed Fourier transform of s_1, then first writing

$$D = e^{\frac{j2\pi\beta}{N}} \qquad (1)$$

The frequency dependent CSPE function can be written as

$$CSPE_w = F_{ws_0} F^*_{ws_1} = (\frac{a}{2})^2 \begin{bmatrix} D^* \left\| F_w(D^n) \right\|^2 \\ +2\mathrm{Re}\left\{ e^{j2b} D F_w(D^n) \otimes F_w^*(D^{-n}) \right\} \\ +D \left\| F_w(D^{-n}) \right\|^2 \end{bmatrix} \qquad (2)$$

The windowed transform requires multiplication of the time domain data by the analysis widow, and thus the resulting transform is the convolution of the transform of the window function, w_f, with the transform of a complex sinusoid. Since the transform of a complex sinusoid is nothing but a pair of delta functions in the positive and negative frequency positions, the result of the convolution is merely a frequency-translated copy of w_f centred at $+\beta$ and $-\beta$. Consequently, with a standard windowing function, the $\left\| F_w(D^n) \right\|$ term is only considerable when $k \approx \beta$, and it decays rapidly when k is far from β. Therefore, the analysis window must be chosen carefully so that it decays rapidly to minimize any spectral leakage into adjacent bins. If this is so it will render the interference terms, i.e. the second and third terms, to be negligible in (2). Thus, the CSPE for the positive frequencies gives:

$$CSPE_w \approx \frac{a^2}{4} \left\| F_w(D^n) \right\|^2 D^{-1} \qquad (3)$$

Finding the angle of (2) leads to the CSPE frequency estimate

$$f_{CSPE_w} = \frac{-N\angle(CSPE_w)}{2\pi} = \frac{-N\angle\left(\frac{a^2}{4}\left\| F_w(D^n)\right\|^2 D^{-1}\right)}{2\pi} = \frac{-N\angle(\frac{a^2}{4}\left\| F_w(D^n)\right\|^2 e^{-j\frac{2\pi}{N}\beta})}{2\pi} = \frac{-N(-\frac{2\pi}{N}\beta)}{2\pi} = \beta \qquad (4)$$

The procedure of the CSPE algorithm is depicted in block diagram form in Figure 1.

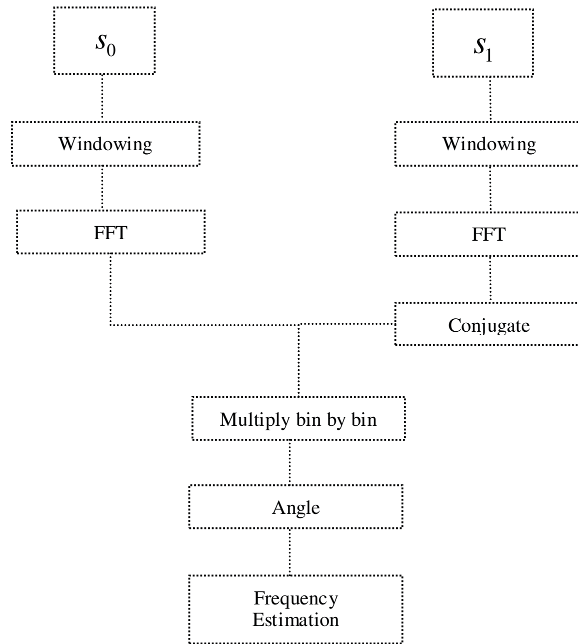

Fig. 1 The flow diagram of CSPE

An example of the output of the CSPE algorithm is shown in Figure 2. Consider the signal S_1 which contains components with frequency values (in Hz) of 17, 293.5, 313.9, 204.6, 153.7, 378 and 423. The sampling frequency is 1024 HZ. A frame of 1024 samples in length is windowed using a Blackman window and is padded using 1024 zeros. The frequency dependent CSPE function is computed as per eq. (2). As shown in Figure 2, each component can be identified exactly and are labelled with an arrow in the graph. The largest error among all the estimates of the components frequencies is approximately 0.15 Hz.

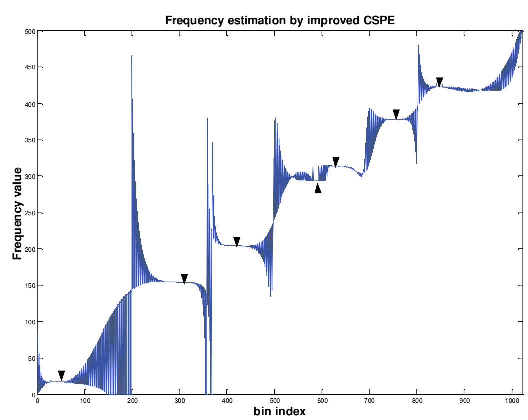

Fig. 2 Frequency estimation of S_1 by CSPE

Notice too in Figure 2 that at the flat sections in the graph of the CSPE result, the width of flat sections where the arrows point are related to the width of the window's main-lobe in the frequency domain.

2.1 Accuracy of the CSPE algorithm

An experiment was carried out to compare the accuracy of the Quadratic Interpolation Estimation Algorithm [4] with the CSPE algorithm. The procedure of this experiment can be described as below: defining twenty centre frequencies f_{c_i} ($0< f_{c_i} < \frac{F_s}{2}$) , for each f_{c_i} , M random frequencies were generated (each of which has a small random fluctuation of f_{c_i}) and thereafter M signals were created based on these M frequencies. The RMS error of the frequency estimation by CSPE and Quadratic Interpolation Estimation Algorithm for these M signals were calculated respectively for each f_{c_i} which shown in the figure below:

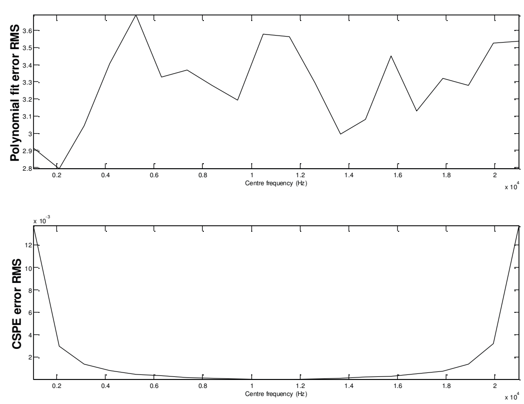

Fig. 3 Accuracy comparison of quadratic fit and CSPE frequency refinements

As shown in figure 3, the CSPE estimate was found to be more accurate than the quadratic interpolation approach, over a factor of 10^3 in many cases.

From the above experiment, it is clear that the CSPE algorithm works very well when the components contained in signal are constant and stable for over the entire data length. However, there can be cases where some components will only appear in half or even a quarter of the data frame length. We can run another experiment on the signal S_2 that has the same frequency components as S_1 , but restricting each component to appear in half or a quarter of the frame. The resulting output of the CSPE algorithm is shown in Figure 4:

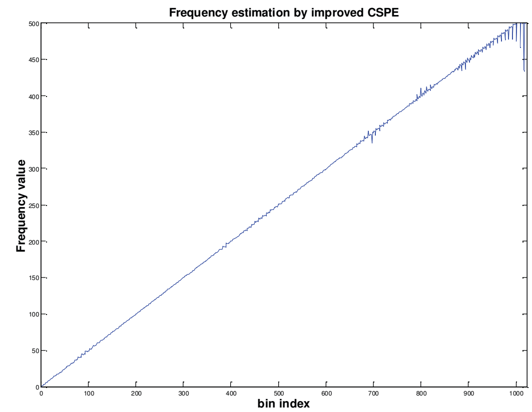

Fig. 4 Frequency estimation of S_2 by CSPE

Frome figure 4, it can be seen that there is no flat region in any part of the graph, that is, none of frequency components can be identified by CSPE algorithm . However, if the FFT magnitude Spectrum of S_2 is plotted, as shown in figure 5, each frequency component is still visible which indicates that there should be some information related to the component present in any FFT-based frequency domain analysis. So, the next section will try to understand this problem and propose a novel approach to deal with it.

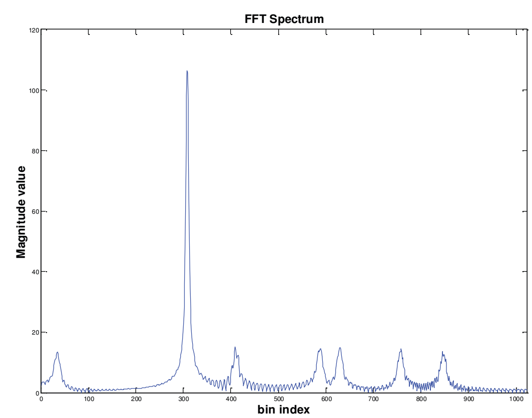

Fig. 5 FFT spectrum of the signal

III. ANALYSIS OF THE PROBLEM AND AN IMPROVEMENT ON THE CSPE ALGORITHM

Let's suppose there are three signals: $x_1[n]$, $x_2[n]$, $x_3[n]$ with the same length 1024 samples, with the same Sampling Frequency 1024 HZ, and all bearing the same frequency component at 123.5 HZ. The difference among the three signals being that the component appears over the entire length of $x_1[n]$, while it only appears in a half and quarter length of signals $x_2[n]$, $x_3[n]$ respectively, the remaining values sample being zero. If we do a normal FFT analysis this component will not be centred on a frequency bin and instead will produce a representation with significant peaks at the 124th and 125th bins with smaller components dying away either side of them. Thus, this signal is an ideal candidate for a CSPE algorithm analysis.

It is possible to rewrite $x_2[n]$ and $x_3[n]$ in terms of the product of $x_1[n]$ and a step function. If $u_1[n]$ and $u_2[n]$ are two different unit step functions then,

$$x_2[n] = x_1[n]u_1[n]; \quad (5)$$

$$x_3[n] = x_1[n]u_2[n]; \quad (6)$$

Denoting $F(x_1)$ and $F(w)$ as the FFT transform of $x_1[n]$ and a suitable window function $w_{ks}[n]$, such as a Blackman window, the spectral representation of the signal $x_1[n]$ can be written as

$$F(x_{1_w}) = F(x_1) * F(w) \quad (7)$$

where * denotes the convolution operator.

Then, the spectral descriptions of the other signals can be written as

$$\begin{aligned} F(x_{2w}) &= F(x_2) * F(w) \\ &= F(x_1) * F(u_1) * F(w) \end{aligned} \quad (8)$$

And likewise,

$$\begin{aligned} F(x_{3w}) &= F(x_3) * F(w) \\ &= F(x_1) * F(u_2) * F(w) \end{aligned} \quad (9)$$

Examining equations (8) and (9) it is possible to interpret the terms $F(u_1) * F(w)$ and $F(u_2) * F(w)$ as the actual windowing operation that are applied to the signal $x_1[n]$ in the frequency domain. Now if we compare the original and alternative window functions frequency response that are all effectively applied to $x_1[n]$ we can see that there is an important difference between the original window and the others in terms of the main lobe size and the height of the side lobes. These are shown in Figure 6.

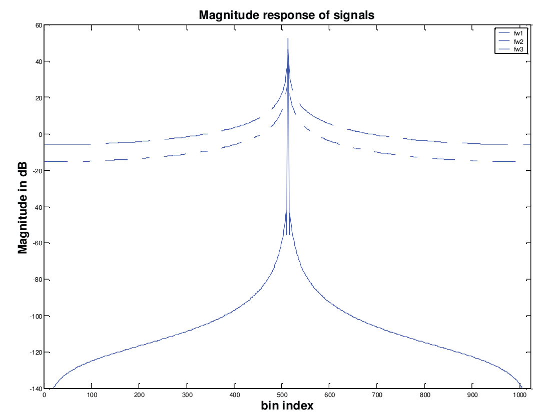

Fig. 6 Magnitude Response of three different actual window functions

From figure 6, when the signal doesn't appear over the entire frame, its actual window function spectrum is significantly different from the original window function. Specifically, the side-lobe hasn't been suppressed in a large extent and the width of main-lobe has been increased. This impacts the CSPE algorithm then in that the interference terms outline in eq. (2) are not sufficiently suppressed. Thus, because these terms are larger, the CSPE output is useless for finding the exact signal frequency of $x_2[n]$ and $x_3[n]$. Motivated by the idea of introducing window function in the first place Fourier analysis, we can introduce a second window function to suppress the greater side-lobes caused by the convolution effect of the spectrum with the unit step functions. This practice is known as apodization [6]. It is more commonly known in image processing than in 1-D signal processing. Normally, an Apodization Function is used to suppress the effects of side-lobes at the expense of lowering the spectral resolution. Some researchers, particularly in image processing [7], [8] have shown that the Kaiser window Function is a better for apodization than other window functions such as the Poisson, Gaussian or Tukey. The Apodization factor using the Kaiser Function can be written as

$$w_{ks}[n] = 1 - kaiser(N, \beta) \quad (10)$$

The side lobe suppression of the Kaiser window is dependent on the parameter β. The apodization of the signal analysis window is then given by

$$w_A[n] = w[n] \left(w_{ks}[n]^{\alpha} \right) \quad (11)$$

Experimentally the relationship between different values of β and effect of raising $w_{ks}[n]$ to an integer power α, to enhance the suppression effects, was evaluated. An example of the effects of suppression of the side-lobes is depicted in Figure 7. From the figure 7, we can find when $w_{ks}[n]$ is raised to a cubic power, α =3, with β = 0.01, it has a side-lobe attenuation level greater than 300 dB.

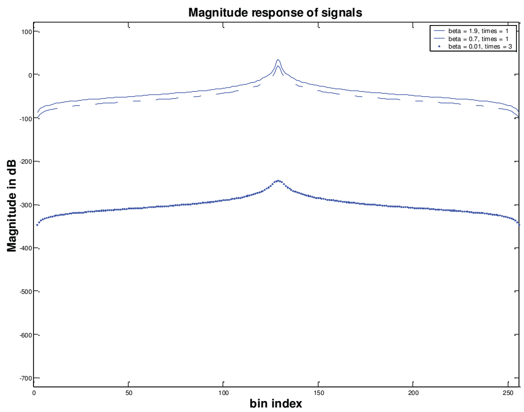

Fig. 7 Magnitude Response of Kaiser window function

Next, the signal S_2 can be analysed with the function $w_A[n]$ (beta = 0.01, α = 3) and the CSPE frequency detection result is shown in Figure 8 where the arrows label the detected frequency components. It can be seen that now the frequency components are identified in the CSPE function.

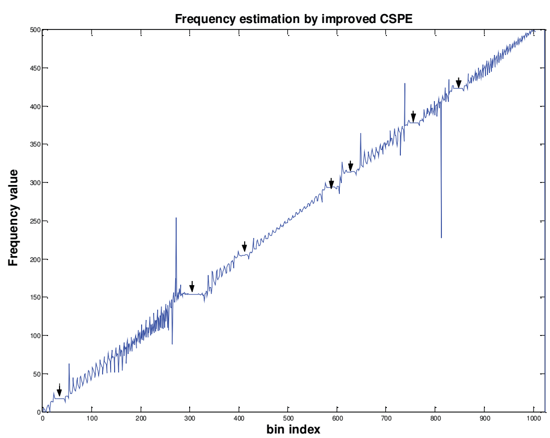

Fig. 8 Frequency estimation by improved CSPE

It was determined experimentally then that when components exist at a different proportion over a frame, the value of β and the power of $w_{ks}[n]$ have to be adapted to get a satisfactory result. Table 1 summarizes this configuration as a reference for users, where Y means a component can be detected, while N means that it cannot; and α means the power of the Kaiser window.

TABLE I

CONFIGURATION OF β AND T FOR THE DETECTION OF DIFFERENT PROPORTION OF A FRAME

	The Proportion of one Frame			
	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
β=0.01, α = 1	Y	N	N	N
β=0.01, α = 3	Y	Y	N	N
β=0.01, α = 10	Y	Y	Y	N
β=0.01, α = 18	Y	Y	Y	Y

IV. CONCLUSIONS AND FUTURE WORK

This paper has addressed a problem discovered with the CSPE algorithm, that is, when frequency component does not exist throughout the entire length of the data frame, that although it appears in the FFT magnitude spectrum, the CSPE algorithm is not capable of detecting this component. By focusing on changing the analysis window's frequency response, the idea of Apodization Function was introduced that was shown to over this difficulty. An experimental result has demonstrated that the performance of the CSPE algorithm has been improved by applying this solution. In future, the intention is to try to extend this CPSE algorithm to correctly identify the dynamic frequency evolution of a frequency modulated signal.

REFERENCES

[1] Florian Keiler, Sylvain Marchand, "Survey on extraction of sinusoids in stationary sounds", Proc. Of the 5 th Int. Conference on Digital Audio Effects(DAFX-02), Hamburg, Germany, September,2002

[2] Short ET AL, "Method and apparatus for compressed chaotic music synthesis", United States Patent 6,137,045; October 2000

[3] Douglas Nelson, Cross Spectral Methods for Processing Speech, Journal of the Acoustic Society of America, vol. 110, No.5, pt.1, Nov.2001, pp.2575-2592

[4] J. Rauhala, H.-M. Lehtonen and V. Välimäki, "Fast automatic inharmonicity estimation algorithm", Journal of the Acoustical Society of America, vol. 121, no. 5, pp. EL184-EL189, 2007.

[5] B. Frei, 'Digital_Sound_Generation – Part 1 Oscillators', Online Book, Institute for Computer Music and Sound Technology, Zurich University of the Arts, Switzerland.. 2007.

[6] H. C. Stankwitz, R. J. Dallaire, and J. R. Fienup, "Non-linear Apodization for Sidelobe Control in SAR Imagery", IEEE Trans. On Aerospace and Elect. Syst., Vol. 31, No. 1, pp. 267-279, Jan. 1995

[7] Thomas, G., Flores, B.C., Jae Sok-Son, "SAR sidelobe apodization using the Kaiser Window", Image Processing, 2000. Proceedings.