NOTES ON MODEL-BASED NON-STATIONARY SINUSOID ESTIMATION METHODS USING DERIVATIVES

Wen Xue and Mark Sandler
Centre for Digital Music, School of EE & CS, Queen Mary, University of London
London, United Kingdom
{xue.wen;mark.sandler}@elec.qmul.ac.uk

ABSTRACT
This paper reviews the derivative method and explores its capacity for estimating time-varying sinusoids of complicated parameter variations. The method is reformulated on a generalized signal model. We show that under certain arrangements the estimation task becomes solving a linear system, whose coefficients can be computed from discrete samples using an integration-by-parts technique. Previous derivative and reassignment methods are shown to be special cases of this generic method. We include a discussion on the continuity criterion of window design for the derivative method. The effectiveness of the method and the window design criterion are confirmed by test results. We also show that, thanks to the generalization, off-model sinusoids can be approximated by the derivative method with a sufficiently flexible model setting.

1. INTRODUCTION

Estimation of parameters of slow-varying sinusoids has received much attention in the digital audio researches, and especially in the area of sinusoid modelling of audio and speech [1]. A slowly-varying sinusoid is described by its instantaneous phase and amplitude functions, i.e.

\[ s(t) = a(t)e^{j\omega(t)} \tag{1} \]

Unlike the case of constant sinusoids, which only have three parameters, for slow-varying sinusoids the parameter set has a higher degree of freedom than the signal itself. Extra information regarding the behaviour of parameters is therefore necessary to reach at any unique set of parameters from the signal.

One widely adopted way to handle this issue is to assume that the sinusoid parameters obey certain parametric model within a short interval from which they are estimated. Early methods assume short-time parameter stationarity and estimate three basic sinusoid parameters, i.e. frequency, amplitude and phase angle, as if they are constant [1][2]. Later methods developed this assumption by involving extra parameters that describe parameter variations. For example, [3] and [4] assume linear frequency and amplitude, while [7] assumes linear frequency and constant amplitude. Although not all sinusoids obey these models, the methods are generally helpful in reducing estimation errors.

Several parameter estimation methods involve taking derivatives or differences regarding time. For example, in [7] the derivatives of the window function are used for calculating spectra, in [8] the instantaneous frequency is estimated by taking the difference of phase angles, while in [9] it is estimated by comparing the spectra of the signal and its derivative. If we look at the simplest case \( s = ae^{ke_j\omega(t)} \), then

\[ s' = j\omega s \tag{2a} \]

By taking the short-time Fourier transform (STFT) of (2a) we get

\[ S'_s(t,\omega) = j\omega S_s(t,\omega) \tag{2b} \]

where the STFT is defined as

\[ S_s(t,\omega) = \int s(t+\tau)w(\tau)e^{-j\omega\tau}d\tau \tag{3} \]

The key feature of (2a) is that by taking the derivative, the parameter \( \omega \) is singled out as a coefficient that applies to \( s \), and remains at a similar position in the frequency-domain equivalent (2b). Once \( S'_s(t,\omega) \) and \( S_s(t,\omega) \) are computed for a proper pair of \( t \) and \( \omega \), the angular frequency \( \omega_0 \) can be solved from (2b). This is the derivative method for constant sinusoids [9]. The conversion into frequency domain serves to keep wide-band or distant-frequency noises from partaking in the calculation.

In this paper we explore the possibilities of the derivative framework in a more general sense, including the existing non-stationary derivative and reassignment methods as special cases. A systematic approach is taken that allows highly complicated signal models, while examples are given to illustrate technical details as appear in various special cases. Section 2 formulates the derivative method for a general signal model as well as several specific ones; section 3 discusses the numeric solution of the general model along with window continuity considerations; section 4 tests both the effectiveness of the method and the continuity criterion regarding window design.

2. THE DERIVATIVE METHOD

We give the generalized signal model of the derivative method as

\[ s(t) = e^{R(t)} \quad R(t) = \sum_{m=0}^{M-1} r_m h_m(t) \tag{4} \]

where \( h_m(t) \) are fixed real functions of \( t \) with sufficient order of continuity, and \( r_m \) is a flexible complex coefficient of \( h_m \). If we separate the real and imaginary parts of \( R \) as

\[ R(t) = P(t) + jQ(t) \quad P(t), Q(t) \in \mathbb{R} \tag{5a} \]

then

\[ P(t) = \sum_{m=0}^{M-1} (\text{Re} r_m) h_m(t) \quad Q(t) = \sum_{m=0}^{M-1} (\text{Im} r_m) h_m(t) \tag{5b} \]

Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09), Como, Italy, September 1-4, 2009

DAFX-1
Taking the derivative of (4) we get
\[ S'(t) = \sum_n r_n S_n(t) h'_n(t) \] (6a)

Applying STFT on (6a) we get
\[ S'_n(t, \omega) = \sum_{t=0}^{M-1} r_n s(t) h'_n(t + \tau) e^{-i\omega \tau} d\tau \] (6b)

(6b) is a linear equation of the coefficients \( r_n, \) \( m = 1, \ldots, M-1. \) Notice that \( r_0 \) is not covered by this equation, since \( h'_n(0) = 0. \) When evaluated for \( t=0 \) (6b) is simplified as
\[ S'_n(\omega) = \sum_n r_n s(t) h'_n(\omega) e^{-i\omega \tau} d\tau, \] (6c)
in which case the coefficient of \( r_n \) can be regarded as the STFT of \( s(t) \) calculated with the window function \( h'_n. \) By (6c) alone we can solve models with up to 2 real parameters apart from \( r_0. \)

Example 2. Applying the exponentially enveloped constant sinusoid model \((M=2, h_0=1, h_1(t)=t, h_2(t)=0.5t^2, p_1=p_2=0)\) in (6c) we get
\[ S'_n(\omega) = \int s(t) w(t) e^{-i\omega \tau} d\tau + q_1 \int s(t) n(t) e^{-i\omega \tau} d\tau \] (8a)
where \( n(t) \) is the STFT of \( n(t) \) calculated with window \( n(t). \) (8a) is the “chirped” version of (2b). The parameters \( q_1 \) (angular frequency) and \( q_2 \) (angular chirp rate) can be estimated as
\[ q_1 = \frac{\text{Re} S'_n(\omega) S'_n(\omega)}{\text{Im} S'_n(\omega) S'_n(\omega)}, \quad q_2 = \frac{\text{Re} S'_n(\omega) S'_n(\omega)}{\text{Im} S'_n(\omega) S'_n(\omega)} \] (8b)

Unlike the reassignment method in [7], by (8b) we are able to jointly estimate the frequency and frequency slope without taking any 2nd-order derivative.

2.2. Second derivative: direct form
Taking the derivative of (6a) we get
\[ S''(t) = 2s(t)R''(t) + 2t s(t)R'(t) \] (9a)

Applying STFT on (9a) we get
\[ S''_n(t, \omega) = \sum_{t=0}^{M-1} r_n s(t) h''_n(t + \tau) e^{-i\omega \tau} d\tau \] (9b)

which, when evaluated at \( t=0, \) becomes
\[ S''_n(\omega) = \sum_{t=0}^{M-1} r_n s(t) h''_n(\omega) e^{-i\omega \tau} d\tau \] (9c)

(9c) is a quadratic equation of the coefficients \( r_n, m=1, \ldots, M-1. \) Again \( r_0 \) is not covered. Using (9c) in conjunction with (6c) we can solve models with up to 4 real parameters apart from \( r_0. \)

Example 4. Applying the exponentially enveloped linear chirp model \((M=3, h_0=1, h_1(t)=t, h_2(t)=0.5t^2, p_2=0)\) in (6c) and (9c) we get
\[ S'_n(\omega) = \int s(t) w(t) e^{-i\omega \tau} d\tau + q_1 \int s(t) n(t) e^{-i\omega \tau} d\tau \] (8a)
where \( S_n(\omega) \) is the STFT of \( s(t) \) calculated with window \( n(t). \) (8a) is the “chirped” version of (2b). The parameters \( q_1 \) (angular frequency) and \( q_2 \) (angular chirp rate) can be estimated as
\[ q_1 = \frac{\text{Re} S'_n(\omega) S'_n(\omega)}{\text{Im} S'_n(\omega) S'_n(\omega)}, \quad q_2 = \frac{\text{Re} S'_n(\omega) S'_n(\omega)}{\text{Im} S'_n(\omega) S'_n(\omega)} \] (8b)
2.3. Second derivative: recursive form

We reconsider the derivative of (6a) in the arrangement of
\[ s'(t) = s(t)R'(t) + s'(t)R(t) \]
which leads to
\[ S_n'(a) = \sum_{n=0}^{M-1} r_n \int \tau \cdot s(\tau)h_n'(\tau)d\tau + \sum_{n=0}^{M-1} r_n s(\tau)h_n'(\tau) \]
(11a)

(11b) is the linear version of (9c). Compared to the latter, it not only avoids quadratic terms, but also has fewer summands.

**Example 5.** Applying the exponentially enveloped linear chirp model \((M=3, h_0=1, h_1(t)=t, h_2(t)=0.5t^2, p_0=0)\) in (11b) we get
\[ S_n'(a) = r_n S_n'(a) + j q_n S_n'(a) + S_n'(a) \]
(12a)

then \(q_2\) can be solved from (10a) and (12a) as
\[ q_2 = \text{Im}\left( \frac{S_n'(a)S_n'(a) - S_n'(a)^2}{S_n'(a) + S_n'(a)} \right) \]
(12b)

With (12b) there is no need to test against extraneous roots. Once \(q_2\) is evaluated \(r_1\) can be calculated by (10d).

**Example 6.** Applying the frequency modulated sinusoid model \((M=4, h_0=1, h_1(t)=t, h_2(t)=\cos(\omega_0 t), h_3(t)=\sin(\omega_0 t), p_0=p_2=p_3=0)\) in (6c) and (11b) we get
\[ -jS_n'(a) = q_n S_n'(a) - q_n a_n S_n'(a) + q_n a_n S_n'(a) \]
(13a)

\[ -jS_n'(a) = q_n S_n'(a) - q_n a_n S_n'(a) + q_n a_n S_n'(a) \]
(13b)

where \(S_n/S_n = (S_n'/S_n')\) are the STFT of \(s'\) calculated with window functions \(w(t) = \cos(\omega_0 t) / \sin(\omega_0 t)\), respectively. (13a) and (13b) provide 4 real linear equations, from which parameters \(q_1, q_1\) can be easily solved.

2.4. Higher-order derivatives

For signal models with more than 4 real parameters apart from \(r_0\) we need to take derivatives of orders higher than 2. For an arbitrary order \(k\), we write
\[ s^{(k)}(t) = \left( s(t)R'(t) \right)^{(k-1)} = \sum_{n=0}^{M-1} r_n \frac{(s(t)h_n'(t))^{(k-1)}}{k!} \]
(14a)

by taking the STFT of which we arrive at
\[ S^{(k)}(a) = \sum_{n=0}^{M-1} r_n \int \tau \cdot s(\tau)h_n'(\tau)^{(k-1)}d\tau \]
(14b)

which, when evaluated at \(t=0\), is reduced to
\[ S^{(k)}(a) = \sum_{n=0}^{M-1} r_n \int \tau \cdot s(\tau)h_n'(\tau)^{(k-1)}d\tau \]
(14c)

(14c) is the order-\(k\) version of (6c) and (11b). Obviously it is a linear equation of the coefficients \(r_n\), or equivalently, two linear equations of \(p_n\) and \(q_m\) (14b) and (14c) are also the general formulations of the (recursive) derivative method. By substituting \(k=1, 2, ..., K\) in (14c) we obtain a system of \(K\) complex (2\(K\) real) linear equations, from which up to 2\(K\) real parameters can be solved.

3. COMPUTATION ISSUES

In this section we explain the numerical implementation of the derivative method proposed in section 2, which can not be applied directly due to the general non-availability of \(s\) and its derivatives in their continuous form. In practice we only have the signal \(s\) in its sampled form
\[ s_n = s(n), n \in \mathbb{Z}, \]
(15)

from which we need to calculate the coefficients of \(r_m\) in (14b) or (14c). All these coefficients have the form of
\[ X_n = \int x^{(k)}(\tau) w(\tau) e^{-j\omega_\tau} d\tau \]
(16a)

where \(x(\tau)\) is known at \(\tau \in \mathbb{Z}\), and \(x^{(k)}\) stands for \((dx/d\tau)^k x\). In equation (14b) \(x(\tau)\) corresponds to \(s(t + \tau)h_n'(t + \tau)\) in (14c) to \(s(t)h_n'(t)\). For numerical computation we assume that \(w\) is compactly supported on \([-T, T]\), \(T \in \mathbb{Z}^+\).

3.1. Derivatives

In the literature of derivative methods attempts have been reported to calculate signal derivatives by interpolating \(s\), e.g. [6]. This, however, risks breaking the signal model, and may require extra data outside the duration of the window function. In this paper we take the approach of integration by parts, which transfers the differentiation operators from the signal \(s\) to the window function \(w\). We extend (16a) by allowing differentiation of the window function \(w\):
\[ X_n^{(k)} = \int x^{(k)}(\tau)w(\tau) e^{-j\omega_\tau} d\tau \]
(16b)

Integrating the right side of (16b) by parts we get
\[ X_n^{(k)} = \int w^{(k)} e^{-j\omega_\tau} d\tau \]
(17a)

where
\[ w^{(k)} = \int w^{(k-1)} e^{-j\omega_\tau} - j\omega w^{(k-1)} e^{-j\omega_\tau} d\tau \]
(17a)

If we let \(w^{(0)}(\tau) = w(\tau)\) then (17a) is simplified as
\[ X_n^{(k)} = X_n^{(k-1)} + j\omega X_n^{(k-1)} \]
(17b)

(17b) reduces the differentiation order of \(s\) by 1. By repeating (17b) recursively we are able to, eventually, calculate (16a) without differentiating \(s\).

**Example 7.** Using (17b) in (8a) for linear chirps we get
\[ -S_n(\omega) + j\omega S_n(\omega) = jq_n S_n(\omega) + q_n S_n(\omega) \]
(18a)

from which we have
\[ q_n + q_n R e S_n(\omega) S_n(\omega) = \omega - \text{Im} S_n(\omega) \]
(18b)

(18b) can be regarded as a reassignment equation [9], where \(\text{Re} S_n(\omega)/S_n(\omega)\) and \(-\text{Im} S_n(\omega)/S_n(\omega)\) are the amounts of time and frequency reassignments, respectively. It provides an alternative proof that reassignment is perfect for linear chirps.

**Example 8.** Using (17b) in (10a) and (12a) for exponentially enveloped linear chirps we get
\[-S_\omega(\omega) + jaoS_\omega(\omega) = r_1S_\omega(\omega) + jq_2S_{\omega^2}(\omega) \]  
\[S_\omega(\omega) - j2aoS_\omega(\omega) - \omega^2 S_{\omega^2}(\omega) = r_1(-S_\omega(\omega) + jaoS_\omega(\omega)) - q_2(jS_{\omega^2}(\omega) + aoS_{\omega^2}(\omega)) \]  
\[\text{From which we eliminate } r_1 \text{ and get} \]
\[q_2 = \frac{\text{Im}S_{\omega^2}(\omega) - \text{Im}S_\omega(\omega)S_{\omega^2}(\omega)}{\text{Re}S_{\omega^2}(\omega)S_\omega(\omega) - \text{Re}S_\omega(\omega)S_{\omega^2}(\omega)} \]

(19c) is the reassignment method [7] for calculating linear chirp rates, which requires 1st and 2nd derivatives. By this example we have shown that the same method applies to linear chirps with exponential amplitudes too. However, the methods in [7] for calculating instantaneous frequency can not be directly applied. One solves (19a) instead.

To implement (14c) for \(k=1, \ldots, K\) we need to calculate \(S_{\omega^{(k)}}(\omega)\) for \(k=1, \ldots, K\) and \(X_{\omega^{(k)}}(\omega)\) for \(\omega=0, \ldots, \omega=1, \ldots, 2\omega-1\), \(M-1\), where \(x_n = s \cdot h_n\). These are evaluated at the same \(\omega_1\) which can be selected at the DFT peak corresponding to the sinusoid. The following chart gives the data flow for calculating \(S_{\omega^{(k)}}(\omega)\), \(k=1, \ldots, K\):

<table>
<thead>
<tr>
<th>(S_\omega)</th>
<th>(S_{\omega'})</th>
<th>(S_{\omega''})</th>
<th>(\ldots)</th>
<th>(S_{\omega^{(k-1)}})</th>
<th>(S_{\omega^{(k)}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\downarrow)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\downarrow)</td>
<td>(\downarrow)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>(S'_{\omega})</td>
<td>(S'_{\omega'})</td>
<td>(S'_{\omega''})</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>(\downarrow)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\downarrow)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>(S''_{\omega})</td>
<td>(S''_{\omega'})</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
<tr>
<td>(\hline)</td>
<td>(\hline)</td>
<td>(\hline)</td>
<td>(\hline)</td>
<td>(\hline)</td>
<td>(\hline)</td>
</tr>
<tr>
<td>(S_{\omega^{(k-1)}})</td>
<td>(S_{\omega^{(k)}})</td>
<td>(\downarrow)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
</tr>
</tbody>
</table>

Figure 1: Data flow for calculating \(S_{\omega^{(k)}}, k=1, \ldots, K\).

The first row of the chart is evaluated directly by Fourier transform (see 3.2). The second row can be evaluated either by (17a) or (17b). The rest \(K-1\) rows can only be evaluated by (17b). \(w\) and its derivatives must be 0 at \(-T\) and \(T\) up to order \(K-2\) (or \(K-1\) if (17a) is to be avoided). The computation of \(X_{\omega^{(k)}}(\cdot)\) is very similar to the above, with one less differentiation.

#### 3.2. Discrete Fourier transform

The first row in Figure 1 contains Fourier transforms in the form of

\[X'_{\omega}(\omega) = \int x(t)e^{-j\omega t}dt \]  
where \(x\) has the form \(sw^{(1)}\) or \(shw^{(1)}\). Here the superscript "c" stands for "continuous". \(x\) is only known in its sampled form \(x_s(x(n)), n \in \mathbb{Z}\). We approximately calculate (20a) by

\[X(\omega) = \sum_{k=0}^{\infty} x_k e^{-j\omega k} \]  
(20b) is evaluated only at the selected angular frequency \(\omega\). The relation between \(X(\omega)\) and \(\lambda(\omega)\) is given by the sampling theorem as

\[X(\omega) - X'(\omega) = \sum_{k=0}^{\infty} X'(\omega - 2k\pi) \]  
(21)

According to (21), for \(\lambda(\omega)\) to be a good approximation of \(X(\omega)\), \(X'\) must decay fast enough to be absolutely integrable. We rewrite (17b) as

\[X_{\omega^{(k)}}(\omega) + X_{\omega^{(k-1)}}(\omega) = jaoX_{\omega^{(k-1)}}(\omega) \]  
(22)

(22) shows that the decay rate of \(X_{\omega^{(k)}}(\cdot)\) is roughly outlined by \(l+k\).

With each increase in \(l+k\), the decay becomes slower by order of \(1\). To guarantee that \(X_{\omega^{(k)}}(\omega)\) is absolutely integrable, \(X_{\omega^{(k)}}(\cdot)\) needs to have a decay rate faster than \(\rho^{(k+1)}\). Accordingly \(Xw\) (and therefore \(w\)) needs to be in \(l+k\) times continuously differentiable. In the context of Figure 1, \(w\) and its derivatives up to order \(K\) are required to remain zero at \(-T\) and \(T\). Apart from global integrability, we also require that \(X^c\) has significant decay at \(\omega = 2\pi\). For slow-varying complex sinusoids this is automatically satisfied if \(\omega\) is selected near the instantaneous frequency, so that \(\omega+2\pi\) are about twice the Nyquist frequency from the sinusoid. For real sinusoids it is desirable that \(\omega\) be not too close to 0 or \(\pi\) to avoid the conjugate partial being picked up in (21).

#### 3.3. More on window functions

Both in 3.1 and 3.2 we have raised the continuity issue of the window function \(w\), requiring \(w\) to be continuously differentiable up to a certain order. While in 3.1 it appears as a technical convenience for adopting the integration-by-parts method, in 3.2 it concerns the accuracy of numerical computation. The requirement raised in 3.2 applies to the derivative methods in general as long as the spectrum is calculated from discrete samples, no matter how the differentiation is implemented. Since a lower level of continuity is demanded in 3.1 than in 3.2, we see that the integration by parts does not introduce extra constraint to \(w\).

When the order of differentiation is high the continuity requirement of \(w\) rules out all the commonly used window functions. Rectangular, Hamming, Bartlett, Gaussian and Kaiser windows are either discontinuous or \(C^0\) functions, therefore should not be used in the derivative method. Hann and Blackman windows are \(C^1\) functions, so that they should not be used in derivative methods involving \(2^{nd}\)-order derivatives. Window functions of higher continuity level are rarely seen in the literature.

Highly continuous window functions can be constructed by multiplying or convolving less continuous windows. The multiplication method aims at directly eliminating discontinuity at the vanishing ends of \(w\), i.e. \(-T\) and \(T\). In fact, if \(w_1\) and \(w_2\) are win-
3.4. Extra equations

Given the general derivative method in (14b) or (14c), we notice that the signal strictly obeys (4) and the numerical computation incurs no error, the least square solution is accurate (i.e. the square error is 0), and is equivalent to the solution obtained by dropping the extra equation.

3.5. Amplitude and phase angle

The coefficient \( r_0 \), which represents global amplification and phase shift, is not involved in the linear system derived from (14b) or (14c). Once \( r_1, \ldots, r_M \) have been estimated, various methods can be engaged in evaluating \( r_0 \), one of which is by comparing the spectra of \( s \) with and without the contribution from \( r_0 \):

\[
e^c = \frac{S_s(\omega)}{\int e^{j\omega t} w(t)e^{-j\omega t}dt}
\]

Again the integral is evaluated from discrete data with \( \omega \) selected near the instantaneous angular frequency.

3.6. Extension to multiple frames

The derivative method discussed above focuses on one data frame only. If the signal model can be assumed to be stable over the span of more than one frame, then it is possible to avoid taking high-order derivatives by applying (14b) to multiple frames with the same unknown parameters.

4. TEST RESULTS

We test the model-based derivative method on synthesized real sinusoids. In the first part of the tests the sinusoids are synthesized on signal model (4), and the derivative method evaluates the parameters using the true model setting. In the second part the parameter variations are not known to the estimator, in which case the derivative method yields an approximate result from within the modelled signal space. For all tests the window length of 1024 is used.

4.1. Estimation with the correct model setting

The performance of sinusoid estimation is evaluated by signal-to-residue ratio (SRR), where the residue is calculated by subtracting a sinusoid frame constructed from the estimated signal model from the original sinusoid. The SRR is formulated as

\[
SRR = \frac{\sum_{n=0}^{M-1} \sum w_n(s_n - \hat{s}_n)}{\sum w_n s_n^2}, \quad \hat{s}_n = e^{j\chi}, \quad \hat{R}(t) = \sum_{n=0}^{M-1} \hat{h}_n(t)
\]

where \( \hat{r}_n, \ldots, M-1 \), are the model parameter estimates, and \( w \) is a Hann window used to emphasize the contributions of both signal and residue at the frame centre.

The first test set contains exponentially enveloped linear chirps (signal model \( M=3 \), \( h_0=1, h_1(t)=t, h_2(t)=0.5t^2 \)), \( p_2=0 \). A total number of 3000 frames are included in this test, with \( p_0=0 \), 10 \( q_0 \) values uniformly selected between 0 and 0.45\( \pi \), 10 \( p_1 \) values between 0 and 0.0045, 10 \( q_1 \) values between 255 and 255.9 bins (1 bin=\( \pi/2 \)), and 10 \( q_2 \) values between 0 and 27 bin/frame (1bin/frame=\( \pi/2 \)). To limit the number of frames, \( p_1 \) and \( q_2 \) do
not vary independently, but appear in three groups: an AM-only group in which \( q_2 = 0 \), an FM-only group in which \( p_1 = 0 \), and an AM-FM group in which \( p_1 \) and \( q_2 \) are paired up in order. This test set is summarized in Table 1. 0dB Gaussian white noise is applied to test performance under noisy environment.

Table 1: Test set 1

<table>
<thead>
<tr>
<th>Group</th>
<th>( l_0, i_i = 0 \ldots 9 )</th>
<th>( p_0 = 0, q_0 = 0.05 \pi l_0, q_1 = 255 + 0.1 i_i(\text{bin}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( i_0 = 0 \ldots 9 )</td>
<td>( p_1 = 0.0005 i_i, q_1 = 0 )</td>
</tr>
<tr>
<td>2</td>
<td>( i_0 = 0 \ldots 9 )</td>
<td>( p_1 = 0, q_1 = 3i_i(\text{bin/frame}) )</td>
</tr>
<tr>
<td>3</td>
<td>( i_0 = 0 \ldots 9 )</td>
<td>( p_1 = 0.0005 i_i, q_1 = 3i_i(\text{bin/frame}) )</td>
</tr>
</tbody>
</table>

\((M=3, h_0 = 1, h_i(t) = 1, h_i(t) = 0.5, p_0 = 0.)\)

The result (marked “D”) is compared against the Abe-Smith QIFFT (quadratically interpolated fast Fourier transform) method [5] which was designed for the same signal model. In this paper the QIFFT method is implemented in both its original version in [5] (“QI”) and an enhanced version in [11] (“EQI”). A Gaussian window is required for both QIFFT methods. Theoretically the derivative method yields very accurate results if the window is used, while the QIFFT methods may suffer moderately from the truncation of the Gaussian window at both ends. In the test we use Hann window in the derivative method. Test results for test set 1 are given in Figure 2. For clean signals both the derivative and enhanced QIFFT methods work very well, with the worst SRR above 70dB. There is a gap between 70~100dB between the two methods, which can be attributed to the truncation of Gaussian window in the QIFFT method. Both methods appear to be susceptible to noise. The derivative and original QIFFT methods show similar results with 0dB noise, while the enhanced QIFFT method does not yield consistent results, partially due to random errors of \( p_1 \) being accumulated in the enhancement step.

![Figure 2: Results for test set 1.](Image)

We run a separate test on the same test set to illustrate the importance of window continuity to the derivative method. We apply the method on clean and noisy data with 4 window functions with different order of continuity, namely Hamming window, the discontinuous, Hann window of C\(^1\), Hann\(^3\) window of C\(^2\), and Hann\(^2\) window of C\(^3\). Results on clean data are given in Figures 3 (a)–(c), which show Hann\(^3\)Hann\(^1\)Hann>Hamming consistently. Results on noisy data are given in (d)–(f), showing very close performance for all window functions (so much so that we are not always able to mark them out in the figures). This indicates that the difference between window functions, as a cause of bias, is overridden by the noise.

![Figure 3: Comparing windows for test set 1.](Image)

The second test set contains sinusoids frequency-modulated by a sinusoidal modulator with frequency \( \omega M \) (signal model \( M = 4 \), \( h_0 = 1 \), \( h_1(t) = 1, h_i(t) = \cos \omega M t, h_i(t) = \sin \omega M t, p_1 = p_2 = p_3 = 0 \)). A total number of 6000 frames are included in this set, with \( \omega M = 0.001 \), \( p_0 = 0 \), 10 \( q_0 \) values uniformly selected between 0 and 0.45\pi, 5 \( q_1 \) values between 255 and 255.8 bins (1 bin = \( \pi/2 \)), 10 modulator angular frequency \( \omega M \) between \( 0.333 \times 10^{-3} \) and \( 6.333 \times 10^{-3} \), 10 modulator amplitudes \( a_M \) from 0 to 27 bins (1 bin = \( \pi/2 \)), and 6 modulator phase angles \( \phi_M \) between 0 and \( \pi/2 \), \( q_M = a_M \cos \phi_M, q_3 = a_M \sin \phi_M \). To limit the number of frames, \( \omega M \) and \( a_M \) do not vary independently, but appear in two groups: in the first group \( \omega M \) is fixed at \( 1 \times 10^{-3} \), in the second group \( \omega M \) is fixed at 9 bins. This test set is summarized in Table 2.

Table 2: Test set 2

<table>
<thead>
<tr>
<th>( i_0 = 0 \ldots 9 )</th>
<th>( i_1 = 0 \ldots 4 )</th>
<th>( p_0 = 0, q_0 = 0.05 \pi l_0, q_1 = 255 + 0.3 i_i(\text{bin}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( i_0 = 0 \ldots 9 )</td>
<td>( \omega M = 10^{-3}, a_M = 3 - i_2(\text{bin}) )</td>
</tr>
<tr>
<td>2</td>
<td>( i_0 = 0 \ldots 9 )</td>
<td>( \omega M = 10^{-3}, i_2 - i_1, 3, a_M = 9(\text{bin}) )</td>
</tr>
<tr>
<td>3</td>
<td>( i_0 = 0 \ldots 9 )</td>
<td>( \phi_M = 0.1 \pi i_1, q_3 = a_M \cos \phi_M, \phi_3 = a_M \sin \phi_M )</td>
</tr>
</tbody>
</table>

\((M=4, h_0 = 1, h_i(t) = 1, h_i(t) = \cos 10^{-3} t, h_i(t) = \sin 10^{-3} t, p_1 = p_2 = p_3 = 0.)\)

Results for the second test set are not compared against another estimator, as there has been none reported to support this signal model. Figure 4 compares test results obtained using the four window functions listed above, with the modulation extent and angular rate as the x axes, respectively. Again we have accurate result when the window function has sufficient order of continuity.

![Figure 4: Comparing windows for test set 2.](Image)
4.2. Approximation of unknown model settings

In the second part of the test we show how the derivative method behaves when applied to sinusoids whose \( R(t) \) model is not known to the estimator. In this case it is generally impossible to select a limited number \( M \) of functions \( h_0, \ldots, h_{M-1} \) so that \( R \) is their linear combination. However, if \( R \) is close enough to the linear space with basis \( \{h_0, \ldots, h_{M-1}\} \), then it is possible to approximate \( R(t) \) by a linear combination of the basis functions. In the following we use a simple example to illustrate this.

In this example we let \( h_a(t)=t^a \), so that the above-mentioned function space is that of polynomials of orders up to \( M-1 \). The ability of polynomials to locally approximate arbitrary smooth functions is proved by Taylor’s theorem. For test signals we use a frequency modulated sinusoid with optional accompanying amplitude modulation:

\[
s(t) = \cos(\omega_0 t + \delta_0 \cos \omega_a t + \delta_1 \sin \omega_a t) \tag{28a}
\]

\[
s_1(t) = (1.5 + \cos(\omega_0 t + 1) \cos(\omega_a t + \delta_0 \cos \omega_a t + \delta_1 \sin \omega_a t) \tag{28b}
\]

where \( \omega_0=0.005 \), \( \omega_a=255 \) bin, \( \delta_0=6 \) bin and \( \delta_1=8 \) bin. In (28a) and (28b) both frequency and amplitude modulators are sinusoids with angular frequency \( \omega_0 \). Obviously neither signal fits into any polynomial frequency-and-log-amplitude model.

With the above settings we run the derivative method with \( M=2, \ldots, 7 \). A Hann window is used in this test. To avoid taking derivatives above \( 3^{rd} \) order, we apply the derivative method at two frames (see 3.6), with the second frame being 256 point (i.e. 25%) shifted from the first, therefore 1280 data points are needed here instead of 1024. Equal numbers of derivatives are taken from both frames. Possible extra equations are handled by the least square approach (see 3.4). For \( s_1 \) we solve the system assuming \( p_n=0, \forall m \); for \( s_2 \) no such restriction is applied.

Figure 5 shows progressively the approximation of the instantaneous frequency of \( s_1 \) by polynomials, achieved by the derivative method. The dotted curve (R) is the true frequency, while the solid curves (Rx) are the estimated approximations. The value \( x \) in “Rx” is the degree of the polynomial used for approximating the instantaneous frequency. The time span in these figures is -512–512. In general, the more polynomials we use, the closer the frequency estimate is to the true frequency. We observed that the approximation is better in right half. This can be attributed to the use of a second frame centred at 256.

Figure 6 shows the approximation progress of the instantaneous frequency (a–f) and amplitude (g–l) of signal \( s_2 \). For amplitude results the \( x \) in “Rx” is the degree of polynomial, used for approximating the logarithmic amplitude, minus 1. The results are similar to those observed in Figure 5, with moderately higher bias due to the doubling of the number of unknown variables.

Figure 5: Approximating frequency modulation

![Figure 5: Approximating frequency modulation](image)

Figure 6: Approximating amplitude and frequency modulation

![Figure 6: Approximating amplitude and frequency modulation](image)

We run a numerical experiment on frequency-modulated sinusoids to test the approximation performance using the polynomial basis. A test set 3, based on test set 2 but with a higher modulation rate in group 1, is used for this purpose. Again the derivative method is implemented to take 2 frames with 25% overlap, and the least-square approach in 3.4 is used to handle possible extra equations. The Hann window is used for all values of \( M \). Results are given in SRR, calculated from the first frames (1024 samples) only.

The approximation results are depicted in Figure 7, where we have marked the curves by the different values of \( M \), from 2 to 7. For comparison we also run the enhanced Abe-Smith method, whose results we plot in dotted curves, and the derivative method using the exact model setting, whose results we mark by “…”.

For both test groups the SRR improves consistently as \( M \) becomes larger. The performance of Abe-Smith method is very close to the derivative method with \( M=3 \), which is very reasonable as for \( M=3 \) the two methods actually have the same signal model. The derivative method with exact model setting, unsurprisingly, has the best performance on both test groups.
5. CONCLUSIONS

In this paper we have reviewed the derivative method for non-stationary sinusoid estimation in a generalized signal model (4). We have shown that by modelling the complex logarithm of the signal as the linear combination of a limited number of basis functions, it is possible to obtain a linear system of the combination weights by taking derivatives. The coefficients of the linear system are short-time Fourier transforms involving \( s \), the basis functions and their derivatives. For the evaluation of these coefficients an integration-by-parts method is applied to transfer the differentiation from \( s \) to the window function, whose derivatives are known by design. The derivative method requires the window function be continuously differentiable up to a certain order, for which we have discussed various ways to obtain highly continuous windows. Tests show that the derivative method yields accurate results for clean sinusoids with the correct model setting, and is able to approximate sinusoids of unknown model type with a sufficiently large polynomial basis.

On the other hand, despite the improved flexibility of (4), the derivative method is still model-based, and therefore suffers from deficiencies typical to all model-based estimators, such as potential overfitting for sinusoids outside the modelled signal space. Modelling error due to these deficiencies can be compensated by engaging non-parametric error control methods, such as the one in [11], which is designed to work with arbitrary estimators.

6. ACKNOWLEDGMENTS

This work was supported by the EPSRC EP/E017614/1 project OMRAS2 (Online Music Recognition and Searching).

7. REFERENCES