Low-Complexity Iterative Sinusoidal Parameter Estimation

Jean-Marc Valin[⋆⋄], Daniel V. Smith[†], Christopher Montgomery[‡⋄], Timothy B. Terriberry[⋄]
[⋆]CSIRO ICT Centre, Australia
[†]CSIRO Tasmanian ICT Centre, Australia
[‡]RedHat Inc., USA
[⋄]Xiph.Org Foundation
{jean-marc.valin,daniel.v.smith}@csiro.au
{xiphmont,tterribe}@xiph.org

***Abstract*— Sinusoidal parameter estimation is a computationally-intensive task, which can pose problems for real-time implementations. In this paper, we propose a low-complexity iterative method for estimating sinusoidal parameters that is based on the linearisation of the model around an initial frequency estimate. We show that for N sinusoids in a frame of length L, the proposed method has a complexity of $O(LN)$, which is significantly less than the matching pursuits method. Furthermore, the proposed method is shown to be more accurate than the matching pursuits and time frequency reassignment methods in our experiments.**

I. Introduction

The sinusoidal model is increasingly being used in signal processing applications such as speech synthesis [1], speech coding [2], and audio coding [3]. Estimating the model parameters often represents a significant fraction of the overall complexity of these applications. However, many real-time applications require a very low-complexity estimation algorithm.

This paper proposes a new procedure based on the linearisation of the model around an initial frequency estimate. Parameters are optimised using an iterative method with fast convergence. We show that for typical configurations, it is over 20 times less complex than matching pursuits [4].

We start by introducing sinusoidal modelling and prior art in Section II. Section III discusses frequency estimation and our proposed linearisation. In Section IV, we present a low-complexity iterative solver for estimating sinusoidal parameters. Results are presented in Section V with a discussion and Section VI concludes this paper.

II. Sinusoidal Parameter Estimation

A general sinusoidal model that considers both amplitude and frequency modulation can be used to approximate a signal $\tilde{x}(t)$ as:

$$\tilde{x}(t) = \sum_{k=1}^{N} A_k(t) \cos\left(\int_0^t \omega_k(t)\, dt + \phi_k\right) \ , \qquad (1)$$

where $A_k(t)$ is the time-varying amplitude, $\omega_k(t)$ is the time-varying frequency and ϕ_k is the initial phase. The model in (1) has limited practical use because it is very complex and has an infinite number of ways to approximate $x(t)$. Using discrete time n and normalised frequencies θ_k over a finite window $h(n)$ yields a simpler model:

$$\tilde{x}(n) = h(n) \sum_{k=1}^{N} \left(A_k + A_k^{'} n\right) \cos(\theta_k n + \phi_k) \ , \qquad (2)$$

where $A^{'}$ is the first time derivative of the amplitude, or even

$$x(n) = h(n) \sum_{k=1}^{N} A_k \cos(\theta_k n + \phi_k) \qquad (3)$$

if we do not want to model amplitude variation within a frame. Although simpler, the models in (2) and (3) are still difficult to estimate because they involve a non-linear optimisation problem.

Several methods exist for estimating sinusoidal parameters. The standard DFT over a rectangular window is limited by both frequency leaking (sidelobes from the rectangular window) and its poor frequency resolution equal to $\pi/L\ rad/s$ for a frame of length L.

By defining an over-complete dictionary of sinusoidal bases, matching pursuits methods [4] make it possible to increase the resolution arbitrarily, while allowing for a window in its basis functions. However, being a greedy algorithm, matching pursuits behaves sub-optimally when the basis functions used are not orthogonal [5], which is usually the case for sinusoids of arbitrary frequency over a finite window length. The orthogonality problem of matching pursuits can be mainly overcome by further non-linear optimisation as in [5]. However this requires a significant increase in complexity (such as $O\left(N^4\right)$ terms).

Another approach is the time frequency (TF) reassignment method, which can be used to improve estimates of frequency localisation within various forms of TF representations [6], including spectrograms [6], [7]. In the case of the spectrogram, phase information from the short time Fourier transform (STFT) is exploited to move energy away from the centre of the frequency bin (t, w) to the centre of gravity of the spectral distribution [6]. Hence, this approach can be used to reduce the inaccuracy of frequency estimation in a quantised TF representation that is reliant upon the temporal resolution of the window. A drawback to this approach is that it is not well suited to noisy signal conditions, as energy becomes concentrated in noise dominated regions [7].

Other work, such as [1], [8] focuses on the estimation of sinusoidal partials in harmonic signals. While these methods generally have a low complexity, they are not applicable to non-harmonic signals.

III. Linearised Model

As another way of obtaining accurate frequency estimation, we propose rewriting the sinusoidal model in (2) as

$$\tilde{x}(n) = h(n) \sum_{k=1}^{N} \left(A_k + nA_k'\right) \cdot \cos\left(\left(\theta_k + \Delta\theta_k\right) n + \phi_k\right) \ , \quad (4)$$

where θ_k is an initial estimate of the frequencies and $\Delta\theta_k$ is an unknown correction to the initial estimate. When both the amplitude modulation parameter A_k' and the frequency correction $\Delta\theta_k$ are small, we show in Appendix I that (4) can be linearised as the sum of four basis functions

$$\tilde{x}(n) = h(n) \sum_{k=1}^{N} c_k \cos\theta_k n + s_k \sin\theta_k n + d_k n \cos\theta_k n + t_k n \sin\theta_k n \ , \quad (5)$$

with

$$c_k = A_k \cos\phi_k \ , \quad (6)$$
$$s_k = -A_k \sin\phi_k \ , \quad (7)$$
$$d_k = A_k' \cos\phi_k - A_k \Delta\theta_k \sin\phi_k \ , \quad (8)$$
$$t_k = -A_k' \sin\phi_k - A_k \Delta\theta_k \cos\phi_k \ . \quad (9)$$

From now on, unless otherwise noted, bold uppercase symbols ($\mathbf{A}$) denote matrices, bold lower case symbols ($\mathbf{a}_i$) denote the columns of the matrix and italic symbols ($a_{i,j}$) denote the elements of the matrix. We can express (5) in matrix form as

$$\tilde{\mathbf{x}} = \mathbf{A}\mathbf{w} \ , \quad (10)$$
$$\mathbf{A} = \left[\mathbf{A}^c, \mathbf{A}^s, \mathbf{A}^d, \mathbf{A}^t\right] \ , \quad (11)$$
$$\mathbf{w} = \left[\mathbf{c}, \mathbf{s}, \mathbf{d}, \mathbf{t}\right]^T \ . \quad (12)$$

where the basis components $\mathbf{A}^c$, $\mathbf{A}^s$, $\mathbf{A}^s$ and $\mathbf{A}^t$ are defined as

$$a_{n,k}^c = h(n) \cos\theta_k n \ , \quad (13)$$
$$a_{n,k}^s = h(n) \sin\theta_k n \ , \quad (14)$$
$$a_{n,k}^d = h(n)\, n \cos\theta_k n \ , \quad (15)$$
$$a_{n,k}^t = h(n)\, n \sin\theta_k n \ . \quad (16)$$

The best fit can then be obtained through a least-square optimisation, by posing

$$\frac{\partial}{\partial \mathbf{w}} \left\|\mathbf{A}\mathbf{w} - \mathbf{x}_h\right\|^2 = 0 \ , \quad (17)$$

where $\mathbf{x}_h$ is the windowed input signal. This leads to the well known solution

$$\mathbf{w} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{x}_h \ . \quad (18)$$

Once all linear parameters in (5) are found, the real sinusoidal parameters can be retrieved by solving the system (6)-(9):

$$A_k = \sqrt{c_k^2 + s_k^2} \ , \quad (19)$$
$$\phi_k = \arg\left(c_k - \jmath s_k\right) \ , \quad (20)$$
$$A_k' = \frac{d_k c_k + s_k t_k}{A_k} \ , \quad (21)$$
$$\Delta\theta_k = \frac{d_k s_k - t_k c_k}{A_k^2} \ . \quad (22)$$

IV. Iterative Solver

Though it is far less computationally demanding than a classic non-linear solver, solving the linear system (18) still requires a great amount of computation. In [8], a method was proposed to reduce that complexity from $O\left(LN^2\right)$ to $O\left(N \log N\right)$, but only for harmonic signals. In this paper, we propose an $O\left(LN\right)$ solution without the restriction to harmonic signals.

Our method uses an iterative solution based on the assumption that matrix A is close to orthogonal, so that

$$\left(\mathbf{A}^T\mathbf{A}\right)^{-1} \approx \operatorname{diag}\left\{\frac{1}{\mathbf{a}_1^T\mathbf{a}_1}, \ldots, \frac{1}{\mathbf{a}_N^T\mathbf{a}_N}\right\} = \mathbf{\Phi} \ . \quad (23)$$

That way, an initial estimate can be computed as

$$\mathbf{w}^{(0)} = \mathbf{\Phi}^{-1} \mathbf{A}^T \mathbf{x}_h \quad (24)$$

and then refined as

$$\begin{aligned}\mathbf{w}^{(i+1)} &= \mathbf{w}^{(i)} + \mathbf{\Phi}^{-1}\mathbf{A}^T\left(\mathbf{x}_h - \tilde{\mathbf{x}}^{(i)}\right)\\ &= \mathbf{w}^{(i)} + \mathbf{\Phi}^{-1}\mathbf{A}^T\left(\mathbf{x}_h - \mathbf{A}\mathbf{w}^{(i)}\right)\ . \qquad (25)\end{aligned}$$

It turns out that the iterative method described in (24)-(25) is strictly equivalent to the Jacobi iterative method. The complexity of the algorithm is reduced to $O(LMN)$, where M is the number of iterations required for acceptable convergence. Unfortunately, while the Jacobi method is generally stable for most matrices $\mathbf{A}$ obtained in practice, convergence is not guaranteed and depends on the actual frequencies θ_k.

A. Gauss-Seidel Method

An alternate to the Jacobi method is the Gauss-Seidel method, which has the main advantage that it is guaranteed to converge in this case because the matrix $\mathbf{A}^T\mathbf{A}$ is a symmetric, positive definite matrix [9]. Because the columns of $\mathbf{A}$ are usually nearly orthogonal, $\mathbf{A}^T\mathbf{A}$ is strongly diagonally dominant and the Gauss-Seidel method converges quickly. The linear system can be expressed as

$$\mathbf{R}\mathbf{w} = \mathbf{b}\ , \qquad (26)$$

where

$$\mathbf{R} = \mathbf{A}^T\mathbf{A}\ , \qquad (27)$$

$$\mathbf{b} = \mathbf{A}^T\mathbf{x}_h\ . \qquad (28)$$

If we assume that matrix $\mathbf{A}$ has been pre-normalised ($\mathbf{a}_k^T\mathbf{a}_k = 1, \forall k$), the Gauss-Seidel algorithm is expressed as

$$\begin{aligned}w_k^{(i+1)} =& b_k - \sum_{j<k} r_{k,j}w_j^{(i+1)} - \sum_{j>k} r_{k,j}w_j^{(i)}\\ =& \mathbf{a}_k^T\mathbf{x}_h - \sum_{j<k}\mathbf{a}_k^T\mathbf{a}_j w_j^{(i+1)} - \sum_{j>k}\mathbf{a}_k^T\mathbf{a}_j w_j^{(i)}\\ =& w_k^{(i)} + \mathbf{a}_k^T\mathbf{x}_h - \sum_{j<k}\mathbf{a}_k^T\mathbf{a}_j w_j^{(i+1)}\\ &- \sum_{j\geq k}\mathbf{a}_k^T\mathbf{a}_j w_j^{(i)}\\ =& w_k^{(i)} + \mathbf{a}_k^T\mathbf{x}_h - \mathbf{a}_k^T\left(\mathbf{A}\tilde{\mathbf{w}}_k{}^{(i+1)}\right)\\ =& w_k^{(i)} + \mathbf{a}_k^T\left(\mathbf{x}_h - \mathbf{A}\tilde{\mathbf{w}}_k{}^{(i+1)}\right)\ , \qquad (29)\end{aligned}$$

where

$$\tilde{\mathbf{w}}_k{}^{(i+1)} = \Big[w_0^{(i+1)}, \ldots, w_{k-1}^{(i+1)}, w_k^{(i)}, \ldots, w_{N-1}^{(i)}\Big]^T\ . \qquad (30)$$

We can further simplify the computation of (29) by noting that only one element of $\tilde{\mathbf{w}}_k{}^{(i+1)}$ changes for each step. We thus have

$$w_k^{(i+1)} = w_k^{(i)} + \mathbf{a}_k^T\mathbf{e}_k^{(i+1)}\ , \qquad (31)$$

Algorithm 1 Iterative linear optimisation

```
Compute basis functions (13)-(16).
$\mathbf{w}^{(0)} \leftarrow \mathbf{0}$
$\mathbf{e} \leftarrow \mathbf{x}_h$
for all iteration $i$=1...M do
    for all sinusoid component $k = 1 \ldots 4N$ do
        $\Delta w_k^{(i)} \leftarrow \mathbf{a}_k^T\mathbf{e}$
        $\mathbf{e} \leftarrow \mathbf{e} - \mathbf{a}_k\Delta w_k^{(i)}$
        $w_k^{(i)} \leftarrow w_k^{(i-1)} + \Delta w_k^{(i)}$
    end for
end for
for all sinusoid $k = 1 \ldots N$ do
    $A_k \leftarrow \sqrt{c_k^2 + s_k^2}$
    $\phi_k \leftarrow \arg\left(c_k - \jmath s_k\right)$
    $A_k' \leftarrow \frac{d_k c_k + s_k t_k}{A_k}$
    $\Delta\theta_k \leftarrow \frac{d_k s_k - t_k c_k}{A_k^2}$
end for
```

where $\mathbf{e}_k^{(i+1)}$ is the current error on the approximation and is computed recursively as

$$\mathbf{e}_k^{(i+1)} = \begin{cases} \mathbf{e}_{k-1}^{(i+1)} - \left(w_{k-1}^{(i+1)} - w_{k-1}^{(i)}\right)\mathbf{a}_{k-1} & , \ k \neq 0 \\ \mathbf{e}_N^{(i)} & , \ k = 0 \end{cases}\ . \qquad (32)$$

The resulting computation is summarised in Algorithm 1. If there is only one iteration, then algorithm 1 is equivalent to a simplified version of the matching pursuits algorithm where the atoms (frequency of the sinusoids) have been pre-defined before the search. From this point of view, the proposed method relaxes the orthogonality assumption made by the matching pursuits method.

The main difference with the Jacobi method is the Gauss-Seidel method includes partial updates of the error term after each extracted sinusoid. The convergence also follows intuitively from the fact that each individual step is an exact projection that is guaranteed to decrease the current error $\mathbf{e}$ — or at worst leave it constant if the optimal solution has been reached. Also, because the error term is updated after each component k, placing the highest-energy terms first speeds up the optimisation. For this reason, we first include the $\cos\theta_k n$ and the $\sin\theta_k n$ terms, followed by the $n\cos\theta_k n$ and the $n\sin\theta_k n$ terms. We have found that this usually reduces the number of iterations required for convergence. The resulting algorithm typically converges in half as many iterations as alternative conjugate gradient techniques, such as LSQR [10], which cannot take advantage of the diagonal dominance of the system.

If in (13)-(16) we (arbitrarily) choose $n = 0$ to lie in the centre of the frame (between sample $L/2$ and sample $L/2+1$ if L is even), the $\mathbf{a}_k^c$ and $\mathbf{a}_k^t$ vectors all have even symmetry, while $\mathbf{a}_k^s$ and $\mathbf{a}_k^d$ all have odd symmetry. This leads to the following orthogonality properties:

$$\langle \mathbf{a}_k^c, \mathbf{a}_k^s \rangle = 0 \ , \tag{33}$$

$$\left\langle \mathbf{a}_k^c, \mathbf{a}_k^d \right\rangle = 0 \ , \tag{34}$$

$$\left\langle \mathbf{a}_k^t, \mathbf{a}_k^s \right\rangle = 0 \ , \tag{35}$$

$$\left\langle \mathbf{a}_k^t, \mathbf{a}_k^d \right\rangle = 0 \ . \tag{36}$$

Because the even and odd bases are orthogonal to each other, we can optimise them separately as

$$[\mathbf{c}, \mathbf{t}]^T = \left(\mathbf{A}^{evenT}\mathbf{A}^{even}\right)^{-1} \mathbf{A}^{evenT}\mathbf{x} \ , \tag{37}$$

$$[\mathbf{d}, \mathbf{s}]^T = \left(\mathbf{A}^{oddT}\mathbf{A}^{odd}\right)^{-1} \mathbf{A}^{oddT}\mathbf{x} \ , \tag{38}$$

$$\mathbf{A}^{even} = \left[\mathbf{A}^c, \mathbf{A}^t\right] \ , \tag{39}$$

$$\mathbf{A}^{odd} = \left[\mathbf{A}^d, \mathbf{A}^s\right] \ . \tag{40}$$

Not only does the orthogonality accelerate convergence, but it allows us to split the error $\mathbf{e}$ into half-length even and odd components, reducing the complexity of each iteration by half.

B. Non-Linear Optimisation

If the initial frequency estimates θ_k^0 are close to the real frequency of the sinusoids θ_k, then the error caused by the linearisation (5) is very small. In this case, Algorithm 1 should result in a value of $\theta_k^0 + \Delta\theta_k$ that is even closer to the real frequencies. However, if the initial estimates deviate significantly from the real values, then it may be useful to restart the optimisation from the new frequency estimates. Repeating the operation several times, we obtain a non-linear iterative solver for A_k, θ_k, $A_k^{'}$ and ϕ_k.

We have found that it is not necessary to wait for Algorithm 1 to converge before updating the frequencies θ_k. We can let both the linear part and the non-linear part of the solution run simultaneously. To do that, we must first subtract the solution of the previous iteration before restarting the linear optimisation.

The non-linear method we propose is detailed in Algorithm 2 and shares some similarities with the Gauss-Newton method [11]. However, because the reparametrisation in (6)-(9) allows updates to A_k, $A^{'}$ and ϕ_k to be incorporated into the linear model immediately when solving the normal equations, convergence is greatly improved compared to a standard Gauss-Newton iteration in the original parameters. Just like Algorithm 1, it is possible to reduce the complexity of Algorithm 2 by half by taking advantage of the even-odd symmetry of the basis functions.

Algorithm 2 Non-linear iterative optimisation

```
$\forall k,\ \theta_k = initial\ frequency\ estimate$
$\forall k,\ [A_k, \phi_k, A_k^{'}] \leftarrow 0$
$\mathbf{w}^{(0)} \leftarrow \mathbf{0}$
$\mathbf{e} \leftarrow \mathbf{x}_h$
for all non-linear iteration $i$=1...M do
    for all sinusoid $k$ do
        $c_k \leftarrow A_k \cos\phi_k$
        $s_k \leftarrow -A_k \sin\phi_k$
        $d_k \leftarrow A_k^{'} \cos\phi_k$
        $t_k \leftarrow -A_k^{'} \sin\phi_k$
    end for
    $\mathbf{e} \leftarrow \mathbf{x} - \mathbf{A}\mathbf{w}^{(i-1)}$ (result of the last iteration with updated frequency)
    for all sinusoid component $k = 1 \dots 4N$ do
        $\Delta w_k^{(i)} \leftarrow \mathbf{a}_k^T\mathbf{e}$
        $\mathbf{e} \leftarrow \mathbf{e} - \mathbf{a}_k\Delta w_k^{(i)}$
        $w_k^{(i)} \leftarrow w_k^{(i-1)} + \Delta w_k^{(i)}$
    end for
    for all sinusoid $k = 1 \dots N$ do
        $A_k \leftarrow \sqrt{c_k^2 + s_k^2}$
        $\phi_k \leftarrow \arg\left(c_k - \jmath s_k\right)$
        $A_k^{'} \leftarrow \frac{d_k c_k + s_k t_k}{A_k}$
        $\theta_k \leftarrow \theta_k + \frac{d_k s_k - t_k c_k}{A_k^2}$
    end for
end for
```

V. Results And Discussion

In this section, we characterise the proposed algorithm and compare it to other sinusoidal parameter estimation algorithms. We attempt to make the comparison as fair as possible despite the fact that the methods we are comparing do not have exactly the same assumptions or output. Both the linear and the non-linear versions of the proposed algorithm are evaluated. For all algorithms, we use a *sine window*

$$h(n) = \cos\pi\frac{n - (L+1)/2}{L} \ , \tag{41}$$

so that the result of applying the window to both the input signal $\mathbf{x}$ and the basis functions $\mathbf{a}_k$ is equivalent to a Hanning analysis window. Unless otherwise noted, we use a frame length $L = 256$.

A. Convergence

We first consider the case of a single amplitude-modulated sinusoid of normalised angular frequency $\theta = 0.1\pi$. We start with an initial frequency estimate of $\theta = 0.095\pi$, which corresponds to an error of slightly more than one period over the 256-sample frames we use. The non-linear optimisation Algorithm 2 is applied with different values of α. The convergence speed in Figure 1 shows that for

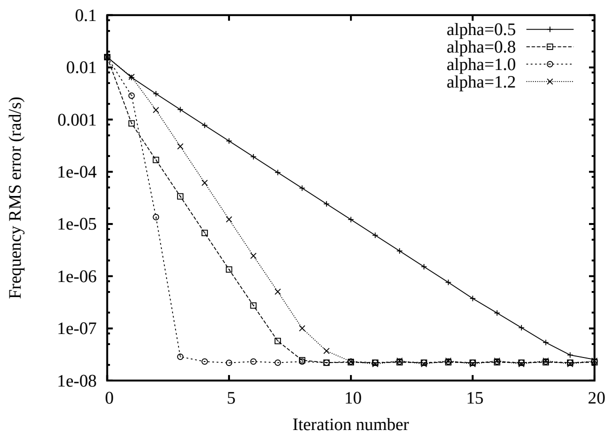

Fig. 1. Convergence of the non-linear optimisation procedure for various values of α. For $\alpha = 1$, convergence is achieved in only 3 iterations. The floor at $2 \times 10^{-8}\, rad/s$ is due to the finite machine precision.

$\alpha = 1$, convergence becomes much faster than for other values of α. This provides a strong indication that the convergence of the algorithm is super-linear, although we give no formal proof.

B. Chirps

We measure the frequency estimation accuracy and the energy of the residual signal for known signals. We use a synthetic signal that is the sum of five chirps with white Gaussian noise. The chirps have linear frequency variations starting at $0.05,\ 0.1,\ 0.15,\ 0.2,\ 0.25\ rad/s$ and ending at $2.0, 2.2, 2.4, 2.6, 2.8 rad/s$, respectively. The relative amplitudes of the chirps are 0 dB, -3 dB, -6 dB, -9 dB, and -12 dB. The following algorithms are considered:

- Time frequency reassignment (**TFR**),
- Matching pursuits (32x over-sampled dictionary) (**MP**),
- Proposed algorithm with linear optimisation (**linear**), and
- Proposed algorithm with non-linear optimisation (**non-linear**).

The time frequency reassignment method is implemented as in [6]. The matching pursuits algorithm uses a dictionary of non-modulated sinusoids with a resolution of $\pi/8192$. We also compare to the theoretical resolution obtained from the picking the highest peaks in the DFT. To make sure that algorithms are compared fairly, all algorithms are constrained to frequencies within one DFT bin of the real frequency, i.e. there are no outliers.

Fig. 2 shows the RMS energy of the residual $(\tilde{\mathbf{x}} - \mathbf{x}_h)$ as a function of the number of iterations for both the linear optimisation and the non-linear optimisation. The linear version converges after only

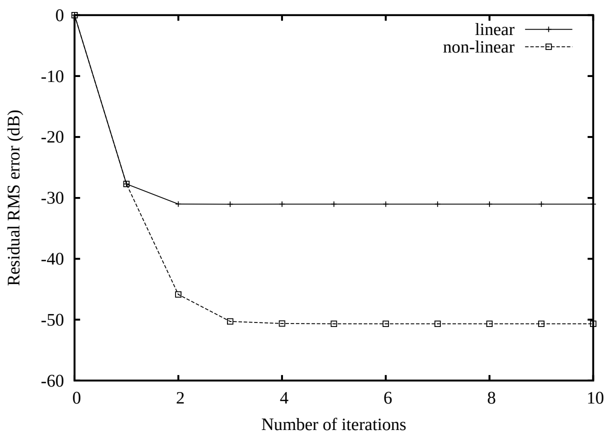

Fig. 2. Reconstruction RMS error as a function of the number of iterations in clean conditions (linear vs. non-linear)

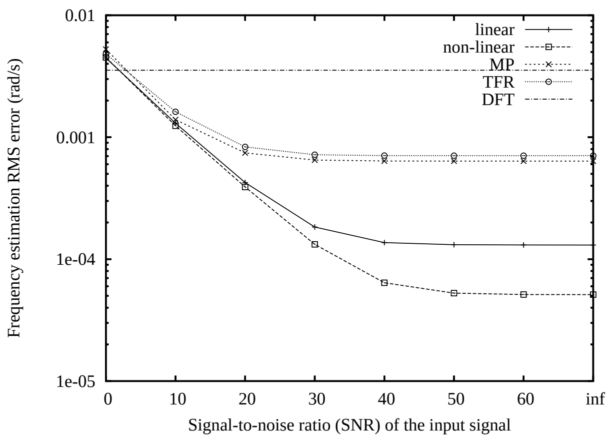

Fig. 3. Frequency RMS estimation error as a function of the SNR.

2 iterations, while the non-linear version requires 3 iterations. We use 3 iterations for both methods in the experiments that follow.

Fig.3 shows the frequency RMS estimation error as a function of the SNR for each of the four algorithms. At very low SNR, all algorithms perform similarly. However, as the SNR increases above 20 dB, matching pursuits stops improving. This is likely due to the fact that the frequencies are not orthogonal, which makes its greedy approach sub-optimal. Both the proposed linear and non-linear approaches provide roughly the same accuracy up to 30 dB, after which the non-linear approach provides superior performance. For this scenario, the only limitation of the non-linear algorithm at infinite SNR is the fact that it does not account for frequency modulation within a frame.

The amplitude estimation error is shown in Fig. 4. Although the behaviour of the amplitude error is similar to that of the frequency estimation error, the difference between the linear and the non-linear optimisation algorithms is accentuated. The time fre-

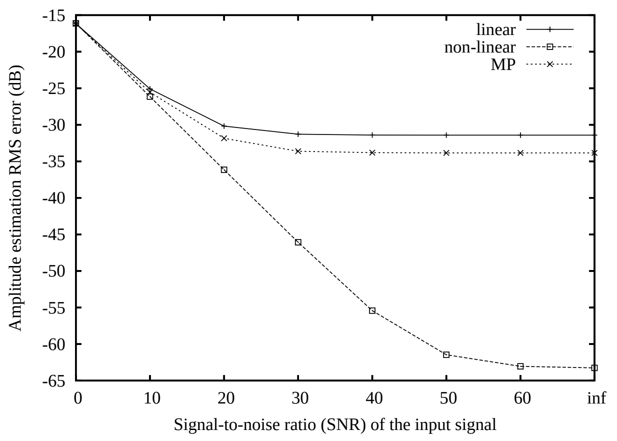

Fig. 4. Amplitude RMS estimation error as a function of the SNR.

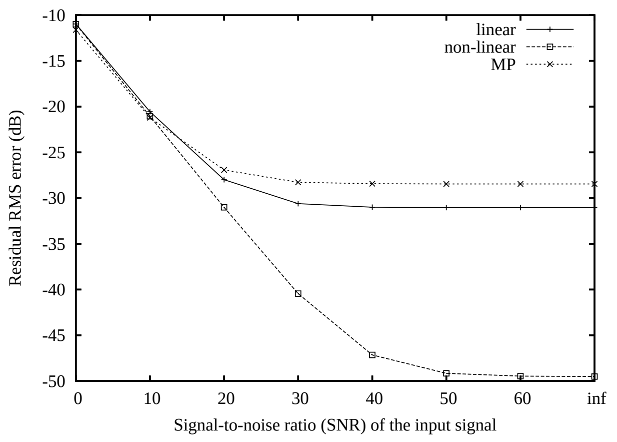

Fig. 5. Reconstruction RMS error as a function of the SNR (the input noise is not considered in the error).

quency reassignment algorithm is not included in the comparison because it does not estimate amplitude.

Fig. 5 compares the reconstruction error for all algorithms, except the time frequency reassignment method, which cannot estimate the amplitude and thus cannot provide a reconstructed signal. The reconstruction error is computed based on the non-noisy version of the chirps. We observe performances similar to the ones in Fig. 3, with the notable exception that when it comes to reconstruction, the non-linear optimisation is able to fit the data much more efficiently than the linear optimisation at high SNR.

We also observe that the performance of our algorithm is slightly worse than that of matching pursuits at low SNR. This can be explained by some slight over-fitting due to the fact that the proposed algorithm also includes an amplitude modulation term. The difference disappears if the amplitude modulation term is forced to zero.

Overall, we observe from the experiment on chirps that our proposed non-linear algorithm clearly out-

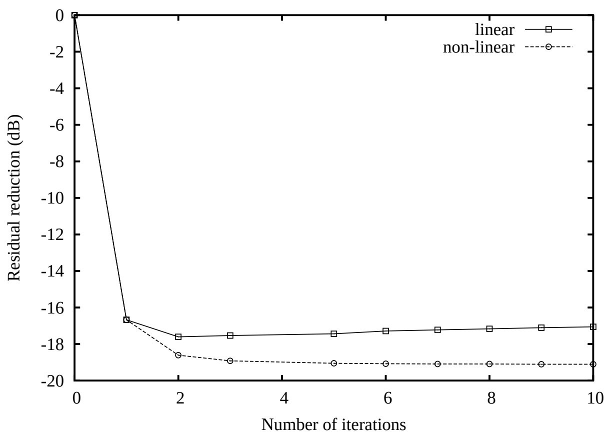

Fig. 6. Reduction in residual energy as a function of the number of iterations.

performs both matching pursuits and time frequency reassignment. The linear version has overall slightly better performance than the other methods, although it does not perform as well as non-linear optimisation. In all cases (Fig. 3 to Fig. 5), all the algorithms compared behave similarly. Their error at low SNR is similar and the slope of the improvement is the same. What differentiates the algorithms is how far they improve with SNR before reaching a plateau.

C. *Audio*

We apply our proposed algorithm to a 90-second collage of diverse music clips sampled at 48 kHz, including percussive, musical, and amusical content. In this case, we cannot compare to the matching pursuits algorithm because the lack of *ground truth* prevents us from forcing a common set of initial sinusoid frequencies. We select the initial frequency estimates required for the proposed algorithm using peaks in the standard DFT.

The energy of the residual is plotted as a function of the number of iterations in Fig. 6. Both algorithms converge quickly and we can see that the linear optimisation only requires 2 iterations, while the non-linear optimisation requires 3 iterations.

D. *Algorithm complexity*

In this section, we compare the complexity of the proposed algorithms to that of other similar algorithms. For the sake of simplicity, we discard some terms that are deemed negligible, e.g., we discard $O(LN)$ terms when $O\left(LN^2\right)$ terms are present.

In Algorithm 1, we can see that each iteration requires $8LN$ multiplications and $8LN$ additions. Additionally, computation of the $4N$ basis functions $\mathbf{a}_k$ prior to the optimisation requires LN additions and $3LN$ multiplications. It is possible to further reduce the complexity of each iteration by taking advantage of the fact that all of our basis functions have either even or odd symmetry. By decomposing the residual into half-length even and odd components, only one of these components needs to be updated for a given basis function. This reduces the complexity of each iteration in Algorithm 1 by half without changing the result. The complexity of each iteration is thus $4LN$ multiplications and $4LN$ additions. For M iterations, this amounts to a total of $(8M+5)\,LN$ operations per frame.

The complexity of the proposed non-linear optimisation algorithm (Algorithm 2) is similar to that of the linear version, with two notable exceptions. First, because the frequency is changing for every iteration, the basis functions need to be re-computed for every iteration. Second, when starting a new iteration, the residual must be updated using the new basis functions. The total complexity is thus $(17M-4)\,LN$ operations per frame (for a single iteration, the linear and non-linear versions are strictly equivalent).

As a comparison a simple matching pursuits algorithm that does not consider modulation requires $4LN^2P$ operations per frame, where P is the oversampling factor (i.e. increase over the standard DFT resolution). If a fast (FFT-based) implementation of the matching pursuits algorithm [5] is used, then the complexity is reduced to $5/2LNP\log_2 LP$.

Table I summarises the complexity of several algorithms. Because the algorithms have different dependencies on all the parameters, we also consider the total complexity in Mflops for real-time estimation of sinusoids in a *typical* scenario, where we have

- frame length: $L=256$,
- number of sinusoids: $N=20$,
- oversampling: $P=32$ (matching pursuits only),
- number of iterations: $M=2$ (linear), $M=3$ (non-linear),
- sampling rate: 48 kHz,
- frame offset: 192 samples (25% overlap).

It is clear from Table I that the proposed algorithms, both linear and nonlinear, reduce the complexity by more than an order of magnitude when compared to matching pursuits algorithms. One must of course take into account that while matching pursuits can estimate the sinusoidal parameters directly from the input signal, the proposed method requires initial frequency estimates.

VI. Conclusion

We have presented a method for estimating sinusoidal parameters with very low complexity. Our proposed method is based on a linearisation of the sinusoidal model, followed by an iterative optimisation of the parameters. The algorithm converges quickly, in only 2 iterations for the linear optimisation and 3 iterations for the non-linear optimisation. It was also shown that the frequency estimation of the non-linear version of our algorithm is more accurate than the matching pursuits and time frequency reassignment methods for the experiment. In addition, we calculated that the complexities of our algorithms were considerably lower than the matching pursuits algorithms.

Like other non-linear optimisation methods, the method we propose requires a good initial estimate of the sinusoids' frequencies. Therefore, low-complexity sinusoid selection is another important problem to investigate for improving sinusoidal parameter estimation. Also, for applications that require it, the proposed algorithm could easily be extended to estimate the frequency modulation within a frame.

Algorithm	Complexity	Typical (Mflops)
MP (slow)	$4LN^2P$	3,300
MP (FFT)	$\frac{5}{2}LNP\log_2 LP$	1,300
linear (18)	$64N^3+32LN^2$	900
non-linear ([5])	$O\left(N^4+LN^2\right)$	>500*
linear (proposed)	$(8M+5)\,LN$	27
non-linear (prop.)	$(17M-4)\,LN$	60

TABLE I
Complexity comparison of various parameter estimation algorithms. *The typical complexity of [5] is not given, but we estimate it to be greater than 500 Mflops.

Appendix I
Linearisation of the Sinusoidal Model

Let us consider a sinusoidal model with piecewise linear amplitude modulation and a frequency offset (from an initial estimate):

$$\tilde{x}(n)=\sum_{k=1}^{N}\left(A_k+nA_k^{'}\right)\cdot \cos\left(\left(\theta_k+\Delta\theta_k\right)n+\phi_k\right)\ , \quad (42)$$

where θ_k is known in advance and $\Delta\theta_k$ is considered small. Using trigonometric identities, we can expand the sum in the cosine term as

$$\begin{aligned}\tilde{x}(n) &= \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\cos\phi_k\cos\left(\theta_k+\Delta\theta_k\right)n \\ &\quad - \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\sin\phi_k\sin\left(\theta_k+\Delta\theta_k\right)n \end{aligned} \tag{43}$$

$$\begin{aligned} &= \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\cos\phi_k\cos\Delta\theta_k n\cos\theta_k n \\ &\quad - \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\cos\phi_k\sin\Delta\theta_k n\sin\theta_k n \\ &\quad - \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\sin\phi_k\cos\Delta\theta_k n\sin\theta_k n \\ &\quad - \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\sin\phi_k\sin\Delta\theta_k n\cos\theta_k n \ . \end{aligned} \tag{44}$$

In the linearisation process, we further assume that $\Delta\theta_k n \ll 1$ and $A_k^{'}n \ll A_k$, so we can neglect all second order terms and above. This translates into the following approximations:

$$\sin\Delta\theta_k n \approx \Delta\theta_k n \ , \tag{45}$$

$$\cos\Delta\theta_k n \approx 1 \ , \tag{46}$$

$$nA_k^{'}\sin\Delta\theta_k n \approx 0 \ . \tag{47}$$

When substituting the above approximations into (44), we obtain:

$$\begin{aligned}\tilde{x}(n) &= \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\cos\phi_k\cos\theta_k n \\ &\quad - \sum_{k=1}^{N}A_k\cos\phi_k\Delta\theta_k n\sin\theta_k n \\ &\quad - \sum_{k=1}^{N}\left(A_k + nA_k^{'}\right)\sin\phi_k\sin\theta_k n \\ &\quad - \sum_{k=1}^{N}A_k\sin\phi_k\Delta\theta_k n\cos\theta_k n \ . \end{aligned} \tag{48}$$

Reordering the terms in (48), leads to the following formulation:

$$\begin{aligned}\tilde{x}(n) &= \sum_{k=1}^{N}A_k\cos\phi_k\cos\theta_k n \\ &\quad - \sum_{k=1}^{N}A_k\sin\phi_k\sin\theta_k n \\ &\quad + \sum_{k=1}^{N}\left(A_k^{'}\cos\phi_k - A_k\Delta\theta_k\sin\phi_k\right)n\cos\theta_k n \\ &\quad - \sum_{k=1}^{N}\left(A_k^{'}\sin\phi_k + A_k\Delta\theta_k\cos\phi_k\right)n\sin\theta_k n \ , \end{aligned} \tag{49}$$

which is a linear combination of four functions. The result in (49) is in fact equivalent to a first-order Taylor expansion. Keeping second order terms would allow us to model both the first derivative of the frequency with respect to time and the second derivative of the amplitude.

References

[1] Y. Stylianou, "Applying the harmonic plus noise model in concatenative speech synthesis," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 1, pp. 21–29, 2001.

[2] P. Hedelin, "A tone oriented voice excited vocoder," in *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, vol. 6, Apr. 1981, pp. 205–208.

[3] S. N. Levine, "Audio representations for data compression and compressed domain processing," Ph.D. dissertation, Stanford University, 1998.

[4] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[5] K. Vos, R. Vafin, R. Heusdens, and W. B. Kleijn, "High-quality consistent analysis-synthesis in sinusoidal coding," in *AES 17th International conference on high quality audio coding*, 2005.

[6] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time scale representations," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1068–1089, 1995.

[7] F. Plante, G. Meyer, and W.Ainsworth, "Improvement of speech spectrograms accuracy by the method of reassignment," *IEEE Trans SAP*, vol. 6, no. 3, pp. 282–287, 1998.

[8] W. D'haes, "A highly optimized method for computing amplitudes over a windowed short time signal: from $O(K^2N)$ to $O(N\log(N))$," in *IEEE Signal Processing Symposium (SPS)*, 2004.

[9] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[10] C. C. Paige and M. A. Saunders, "Lsqr: An algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71, Mar. 1982.

[11] P. E. Frandsen, K. Jonasson, H. B. Nielsen, and O. Tingleff, *Unconstrained Optimization*, 3rd ed. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004.