Fast Least-Squares Solution for Sinusoidal, Harmonic and Quasi-Harmonic Models

Georgios Tzedakis[1], *Yannis Pantazis*[1], *Olivier Rosec*[2] *and Yannis Stylianou*[1]

[1]Institute of Computer Science, FORTH, and Multimedia Informatics Lab, CSD, UoC, Greece
[2]Orange Labs TECH/ASAP/VOICE, Lannion, France
{gtzedaki, pantazis, yannis}@csd.uoc.gr and olivier.rosec@orange-ftgroup.com

10.21437/Interspeech.2010-489

Abstract

Sinusoidal model and its variants are commonly used in speech processing. In the literature, there are various methods for the estimation of the unknown parameters of sinusoidal model such as Fourier transform based on FFT algorithm and Least Squares (LS) method. Least Squares method is more accurate and actually optimum for Gaussian noise, thus, more appropriate for high-quality signal processing, however, it is slower compared with FFT-based algorithms. In this paper, we study the source of computational load of LS solution and propose various computational improvements. We show that the complexity of LS solution as well the execution time are highly improved.

Index Terms: Sinusoidal Models, Harmonic models, Least Squares Method

1. Introduction

Voiced speech is a quasi-periodic signal which is usually modeled in a frame-based manner as a superposition of sinusoids. Sinusoidal modeling based on Fourier transform [1], [2] estimates the sinusoidal parameters by peak picking of the Fourier spectrum using the FFT algorithm. This approach tries to balance between the accuracy of the estimated parameters, their fast computation and the nonstationarity of the analyzed signal. Thus, large windows are necessary for less sidelobe interference between the sinusoids but, then, the local stationarity assumption becomes less valid. Another solution for the estimation of sinusoidal parameters is the Least Squares (LS) method [3], [4] which provides a more accurate solution due to the fact that it tackles the problem of sidelode interference between the components. Hence, shorter analysis windows can be used. However, LS method is much slower compared with FFT algorithm.

In this paper, we address the problem of reducing the computational load of LS solution by doing some of the necessary computations by hand. Fortunately, this can be done due to the fact that sums of complex exponentials can be handled easily as sums of geometric series. Hence, the complexity of the LS solution is improved and the time needed for the computation is reduced. Note also that in the literature there are algorithms which try to reduce the computational costs using lookup tables [5], [6]. However, these methods are especially applied to harmonic model, thus, they are restricted.

The organization of the paper is as follows. Section 2 introduces the sinusoidal, harmonic and quasi-harmonic models whose parameters should be computed. Section 3 presents the computational enhancements we propose while Section 4 shows the improvements of the proposed method in complexity and in CPU time. Finally, Section 5 concludes the paper.

2. Sinusoidal Models

Two models are introduced in this section, namely, the sinusoidal model and an extension of it which we call it quasi-harmonic model [7]. Also, we further distinguish these models depending on their feasible frequency values.

2.1. Sinusoidal Model

Assume that $\mathbf{s} = [s[-N], ..., s[N]]^T$ is the original signal of duration $2N + 1$ samples to be modeled. Sinusoidal model is given by

$$h_0[n] = \sum_{k=-K}^{K} a_k e^{j2\pi f_k n/f_s}, \;\; n = -N, ..., N \quad (1)$$

where K is the number of components, f_k and a_k are the frequency and the complex amplitude of the kth component, respectively, while f_s is the sampling frequency. We are interested in the estimation of complex amplitudes given the number of components and the frequencies, thus, we assume these values known. Note also that when the frequency values are integer multiples of a fundamental frequency, the model is called Harmonic model.

To proceed, Least Squares method minimizes the sum of squared and windowed difference between the original signal, $\mathbf{s}$, and the sinusoidal model, $\mathbf{h_0}$ with respect to a_k. This minimization is reduced to a linear problem hence the LS solution for the complex amplitude is given

26 – 30 September 2010, Makuhari, Chiba, Japan

in matrix form by

$$\mathbf{a} = (E_0^H W^H W E_0)^{-1} E_0^H W^H W \mathbf{s} = R_0^{-1}\mathbf{s_0} \quad (2)$$

where $\mathbf{a} = [a_{-K}, ..., a_K]^T$ is the vector with the complex amplitudes, W is a diagonal matrix with elements the values of the analysis window and matrix E_0 with dimension $(2N+1)\times(2K+1)$ was elements $(E_0)_{nk} = e^{j2\pi f_k n/f_s}$. The complexity of this computation is $O(K^2(N+K))$ and it is quite costly compared with FFT algorithm which is $O(N \log N)$. In this paper, we intend to speed-up the above computation by several ways. Firstly, the computation of $R_0 = E_0^H W^H W E_0$ is improved and, secondly, faster computation of $\mathbf{s_0} = E_0^H W^H W \mathbf{s}$ is proposed. Finally, the inversion of R_0 is studied.

2.2. Quasi-Harmonic Model

The second model which is called quasi-harmonic model [7] has an additional term for each component and it is given by

$$h_1[n] = \sum_{k=-K}^{K} (a_k + n b_k) e^{j2\pi f_k n/f_s}, \quad n = -N, ..., N \quad (3)$$

where b_k is the complex slope for the kth component. Similar to the sinusoidal model there are two possible options for the frequency values. In this case, there is no FFT-based algorithm which is able to compute the parameters of the model, thus, LS is the only option. LS solution for the complex amplitudes and complex slopes is again linear and it is given by

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = ([E_0|E_1]^H W^H W [E_0|E_1])^{-1} [E_0|E_1]^H W^H W \mathbf{s} = \begin{pmatrix} R_0 & R_1 \\ R_1^H & R_2 \end{pmatrix}^{-1} \begin{bmatrix} \mathbf{s_0} \\ \mathbf{s_1} \end{bmatrix} \quad (4)$$

where $\mathbf{a}, W, E_0, R_0$ and s_0 as before, while, $\mathbf{b} = [b_{-K}, ..., b_K]^T$ is the vector with complex slopes, $(E_1)_{nk} = n e^{j2\pi f_k n/f_s} = n(E_0)_{nk}$ is the matrix with elements the complex exponentials multiplied with time. Obviously, the rest matrices are given by $R_1 = E_0^H W^H W E_1$, $R_2 = E_1^H W^H W E_1$ and $\mathbf{s_1} = E_1^H W^H W \mathbf{s}$.

3. Computations

Obviously, the sinusoidal model is a subcase of the quasi-harmonic model, hence, we concentrate only to the computation of the parameters of the quasi-harmonic model.

3.1. Step 1: Fast matrix multiplication

The elements of the submatrices R_0, R_1 and R_2 are given by

$$(R_m)_{ik} = \sum_{n=-N}^{N} w^2[n] n^m e^{j2\pi(f_k - f_i)n/f_s}, \quad m = 0, 1, 2 \quad (5)$$

In order to do the summation of the above equations, we have to consider what kind of window is used. Typically, Hamming, Hanning and rectangular windows are used. These windows are parametrized into a general class of window types given by

$$w_a[n] = (1-a) + a cos(\pi n/N) \quad n = -N, ..., N-1, N \quad (6)$$

Table 1 shows the relationship between various windows and parameter a.

$a = 0$	Rectangular
$a = 0.5$	Hanning
$a = 0.46$	Hamming

Table 1: Different values of a gives various windows.

As (5) asserts, the squared window is also necessary, thus

$$w_a^2[n] = ((1-a) + a cos(\pi n/N))^2 = \\ = d_0 + d_1(e^{j\pi n/N} + e^{-j\pi n/N}) + d_2(e^{j2\pi n/N} + e^{-j2\pi n/N}) \quad (7)$$

where $d_0 = (1-a)^2 + a^2/2$, $d_1 = a(1-a)$ and $d_2 = a^2/4$ are the coefficients.

Applying the squared window (7) to (5), we obtain

$$\begin{aligned} (R_m)_{ik} &= d_0 \sum_{n=-N}^{N} n^m \left[e^{j2\pi(f_k - f_i)/f_s}\right]^n \\ &+ d_1 \sum_{n=-N}^{N} n^m \left[e^{j2\pi(f_k - f_i + \frac{f_s}{2N})/f_s}\right]^n \\ &+ d_1 \sum_{n=-N}^{N} n^m \left[e^{j2\pi(f_k - f_i - \frac{f_s}{2N})/f_s}\right]^n \\ &+ d_2 \sum_{n=-N}^{N} n^m \left[e^{j2\pi(f_k - f_i + \frac{f_s}{N})/f_s}\right]^n \\ &+ d_2 \sum_{n=-N}^{N} n^m \left[e^{j2\pi(f_k - f_i - \frac{f_s}{N})/f_s}\right]^n \end{aligned} \quad (8)$$

To proceed, standard mathematical identity about the sum of geometric series gives that $\sum_{n=0}^{N} \alpha^{\lambda n} = \frac{1-\alpha^{\lambda(N+1)}}{1-\alpha^{\lambda}}$. Taking the derivative with respect of λ, the elements of R_1 show up, thus, they can be computed without performing the summation. Similarly, taking one more derivative, the elements of R_2 are computed. Thus, the elements of the matrices R_i are given by

$$\begin{aligned} (R_m)_{ik} &= d_0 g_m \left(2\pi(f_k - f_i)/f_s\right) \\ &+ d_1 g_m \left(2\pi(f_k - f_i + \frac{f_s}{2N})/f_s\right) + d_1 g_m \left(2\pi(f_k - f_i - \frac{f_s}{2N})/f_s\right) \\ &+ d_2 g_m \left(2\pi(f_k - f_i + \frac{f_s}{N})/f_s\right) + d_2 g_m \left(2\pi(f_k - f_i - \frac{f_s}{N})/f_s\right) \end{aligned} \quad (9)$$

where the auxiliary functions $g_0(x)$, $g_1(x)$ and $g_2(x)$ are given by (10), (11) and (12), respectively.

$$g_0(x) = \begin{cases} \frac{\sin((2N+1)x/2)}{\sin(x/2)}, & x \neq 0 \\ 2N+1, & x = 0 \end{cases} \tag{10}$$

$$g_1(x) = \begin{cases} j\frac{\sin(Nx)}{2\sin^2(x/2)} - jN\frac{\cos((2N+1)x/2)}{\sin(x/2)}, & x \neq 0 \\ 0, & x = 0 \end{cases} \tag{11}$$

$$g_2(x) = \begin{cases} \frac{N^2\cos((N+1)x)+(N+1)^2\cos(Nx)}{2\sin^2(x/2)} - \frac{\sin((2N+1)x/2)}{2\sin^3(x/2)}, & x \neq 0 \\ N(N+1)(2N+1)/3, & x = 0 \end{cases} \tag{12}$$

Finally, due to the fact that the computations of trigonometric functions are expensive, the computation of (9) can be sped up by considering the following identities

$$\cos(\theta+\delta) = \cos(\theta) - [\alpha\cos(\theta) - \beta\sin(\theta)] \tag{13}$$

$$\sin(\theta+\delta) = \sin(\theta) - [\alpha\sin(\theta) - \beta\cos(\theta)] \tag{14}$$

where $\alpha = 2\sin^2(\delta/2)$ and $\beta = \sin(\delta)$ are precomputed coefficients. Thus, the sines and cosines of one of the five terms in (9) is required and the remaining terms are computed using the above formulas.

3.2. Step 2: Faster computation of E_0

We explore, now, a faster way to compute the array E_0 which directly leads to a faster way to compute $\mathbf{s_0}$ and $\mathbf{s_1}$. Again, the most time-consuming part of the computation is the estimation of sines and cosines since the elements of E_0 equals to $(E_0)_{kn} = e^{j2\pi f_k n/f_s} = \cos(2\pi f_k n/f_s) + j\sin(2\pi f_k n/f_s)$. Obviously, having computed matrix E_0, the elements of E_1 are given by $(E_1)_{nk} = n(E_0)_{nk}$. The computational speedup stems from the fact that the solution $c_i(t)$ of the following second-order difference equation

$$c_i(n) - 2\cos(2\pi f_i/f_s)c_i(n-1) + c_i(n-2) = 0, \quad n = 3, 4, \ldots \tag{15}$$

with initial conditions

$$c_i(1) = \cos(2\pi f_i/f_s), \quad c_i(2) = \cos(4\pi f_i/f_s) \tag{16}$$

is given by

$$c_i(n) = \cos(2\pi f_i n/f_s), \qquad n = 1, 2, \ldots \tag{17}$$

Using $c_i(1) = \sin(2\pi f_i/f_s)$ and $c_i(2) = \sin(4\pi f_i/f_s)$ as initial conditions, we produce $c_i(n) = \sin(2\pi f_i n/f_s)$. Thus, using the above equations, the computation of each trigonometric function is replaced by a multiplication.

3.3. Step 3: Matrix Inversion

For the special case where the frequency values are integer multiples of a fundamental frequency the matrices R_0, R_1 and R_2 are Toeplitz. This observation leads into two improvements. Firstly, the construction of $R_m,\ m = 0, 1, 2$ requires only the computation of $K+1$ elements and, secondly, Levinson-type algorithms can be applied [8].

Up to now, no approximation or discretization was performed and the LS solution has no additional error. However, if we allow for small errors, we can achieve faster inversion for the matrix by discarding elements away from the diagonal. This approximation is valid because sinusoids which are away from each other has little or no interference. Thus, if we keep K_0 diagonals, the inversion is sped up significantly.

4. Evaluation

4.1. Complexity

Before Step 1, computing $R_m,\ m = 0, 1, 2$ has computational cost $O(K^2N)$ considering that W is diagonal. After Step 1 each element of R_m is computed in constant time, thus, reducing the complexity to $O(K^2)$. With Step 2, we achieve the computation of the sequences $\sin(2\pi f_k n/f_s), \cos(2\pi f_k n/f_s)$ using approximately $8NK$ multiplications (2 multiplications per element of E_0). Generally, the complexity of the matrix inversion is $O(K^3)$ however, the harmonic model, the inversion of R_0 has complexity $O(K^2)$ due to the Toeplitz structure of the inverted matrix. In the case where the approximation is performed, the complexity is reduced to $O(K_0^2K)$. Totally, the complexity was reduced from $O(K^2(N+K))$ to $O(K(K_0^2+N))$.

4.2. CPU Time

For all the experiments we created synthetic signals as

$$s[n] = \sum_{k=0}^{K} \cos(2\pi f_k n/f_s),\ n = -N, ..., N$$

where $f_s = 16000$ which is a typical value for the sampling frequency. Parameters K and N take values $K = 10, 20, \ldots, 50$ and $N = 150, 175, \ldots, 250$, respectively. For the special case of harmonic frequencies, $f_k = kf_0, k = 0, \ldots K$, fundamental frequency, f_0, is chosen uniformly from the interval $(85-255\text{Hz})$. For the general case when frequencies f_k have no relation to one another, we set $f_0 = 0$ and the rest of the frequencies were uniformly chosen from the interval $\left(85 - \frac{f_s}{2}\text{Hz}\right)$ under the conditions that every two frequencies should be at least 85Hz apart and that $f_{k-1} < f_k$. Each one of the improvements in computational cost was run 1000 times.

The computer used for the experiments was equipped with: Intel Core 2 6600 CPU @ 2.4 GHz and 2GB RAM. Note that only one CPU was used to ensure accuracy of the results. The operating systems was Windows XP Professional 32 bit.

In Tables 2 and 3 can be seen the average CPU time it took to execute each improvement. Note that Step 0 is the case where no improvements are present while Step 1 is the case where $R_m,\ m = 0, 1, 2$ are quickly computed using the functions $g_0(x),\ g_1(x),\ g_2(x)$ and (13)-(14). Step 2 has the additional improvement of faster computation of E_0 by using the difference equation (15), as well as Step 1 is included. Finally, Step 3-K_0 has Steps 1 and 2 incorporated plus the fact that R_0 is K_0-diagonalized. Note that at Table 3 the Signal-to-Noise Ratio (SNR) is also reported.

We observe that from Step 0 to Step 2 the improvement for the kf_0 case is 70% and 60% for the SM (Sinusoidal Model) and the QHM (Quasi-Harmonic Model) respectively while the improvement for the f_k case is 32% and 30%, respectively. The reason for this difference in performance stems from the fact that matrix R_0 is Toeplitz in the harmonic case. The least improvement can be spotted from Step 0 to Step 1 for SM and f_k case, less than 1%. The Steps 3-K_0 are always around 70-80% faster with no significant loss of SNR from turning R_0 into a band matrix.

	SM		QHM	
	kf_0	f_k	kf_0	f_k
Step 0	4.149 ms	4.213 ms	10.734 ms	10.798 ms
Step 1	2.547 ms	4.118 ms	5.282 ms	8.463 ms
Step 2	1.276 ms	2.881 ms	4.228 ms	7.608 ms

Table 2: Average CPU time for improvements 1,2.

	SM			
	CPU time		SNR	
	kf_0	f_k	kf_0	f_k
Step 0	4.149 ms	4.213 ms	279 dB	274 dB
Step 3-3	0.918 ms	1.135 ms	88 dB	84 dB
Step 3-5	0.943 ms	1.059 ms	106 dB	110 dB
Step 3-7	0.967 ms	1.143 ms	118 dB	125 dB

Table 3: Average CPU time and SNR for additional improvements of Step 3.

5. Conclusion and Future work

In the context of sinusoidal representation, we present computational improvements of LS method for the estimation of unknown complex parameters. Exploiting the fact that some of the computations can be performed by hand we speedup the estimation process. The improvements are important especially for the case where the frequency values were integer multiples of a fundamental frequency. Future plans involve the use of lookup tables and further speedup the computation of QHM parameters.

6. Acknowledgement

This work was partially supported by LISTA. The project LISTA acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 25623

7. References

[1] R. J. McAulay and T. F. Quatieri. Speech Analysis/Synthesis based on a Sinusoidal Representation. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34:744–754, 1986.

[2] S. M. Kay. *Modern Spectral Estimation: Theory and Applications*. Prentise-Hall, Englewood Cliffs, NJ, 1988.

[3] T. Kailath. *Linear Least-Squares Estimation*. Halsted Press, 1977.

[4] Y. Stylianou. *Harmonic plus Noise Models for Speech, combined with Statistical Methods, for Speech and Speaker Modification*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1996.

[5] M. Vasilakis, Y. Agiomyrgiannakis, and Y. Stylianou. Fast Analysis/Synthesis of Harmonic Signals. In *ICASSP*, pages III 1004–III 1007, 2006.

[6] Y. Stylianou. A Simple and Fast Way of Generating a Harmonic Signal. *IEEE Signal Processing Letters*, 7:111–113, May 2000.

[7] Y. Pantazis, O. Rosec, and Y. Stylianou. On the Properties of a Time-Varying Quasi-Harmonic Model of Speech. In *Interspeech*, pages 1044–1047, Brisbane, Sep 2008.

[8] P. Delsarte and Y. Genin. An Extension of the Split Levinson Algorithm and its Relatives to the Joint Process Estimation Problem. *IEEE Trans. Inform. Theory*, 34:482–485, 1989.