SURVEY ON EXTRACTION OF SINUSOIDS IN STATIONARY SOUNDS

Florian Keiler[1] *and Sylvain Marchand*[2]

[1]University of Federal Armed Forces, Holstenhofweg 85, D-22043 Hamburg, Germany
[2]LaBRI, University of Bordeaux 1, 351 cours de la Libération, F-33405 Talence cedex, France
florian.keiler@unibw-hamburg.de, sm@labri.u-bordeaux.fr

ABSTRACT

This paper makes a survey of the numerous analysis methods proposed in order to extract the frequency, amplitude, and phase of sinusoidal components from stationary sounds, which is of great interest for spectral modeling, digital audio effects, or pitch tracking for instance. We consider different methods that improve the frequency resolution of a plain FFT. We compare the accuracies in frequency and amplitude of all these methods. As the results show, all considered methods have a great advantage over the plain FFT.

1. INTRODUCTION

Several methods have been proposed in order to extract sinusoidal components by improving the frequency resolution of a plain FFT. With these methods it is possible to accurately recover the frequency, amplitude, and possibly phase of all harmonic components of the sound.

We describe in Section 2 six methods for extracting sinusoidal components from stationary sounds. We show in Section 3 how it is possible to improve these methods even further. Finally, we make a comparison of these analysis methods in Section 4 and we discuss the results of several important tests.

2. DESCRIPTION OF THE METHODS

To improve the frequency resolution of the FFT, different methods are reported in the literature. The methods use different approaches for the frequency analysis of sound signals. In the following subsections we give brief descriptions of the used methods. Table 1 shows an overview of the methods with the window functions and method numbers used in the following sections.

No.	Method	window
1	plain FFT	Hann
2	parabolic interpolation	parabolic main lobe (dB) in freq. domain
3	triangle algorithm	triangle in freq. domain
4	spectral reassignment	Hann
5	derivative algorithm	Hann
6	phase vocoder	Hann

Table 1: *Overview of the different methods and used window functions.*

2.1. Plain FFT without Post-processing

The input frame $s(n)$ is windowed by $x(n) = s(n) \cdot w(n)$ with a window function $w(n)$. We use a Hann window[1] before performing the FFT. Then the FFT $X(k)$ of the windowed input frame is calculated. The FFT magnitude $X_m(k) = |X(k)|$ is set to zero if it is below a small threshold. The modified FFT magnitude is searched for local maxima. For all detected maxima k_m we calculate the estimated sine frequencies and amplitudes as

$$\hat{f}_0 = k_m \frac{f_S}{N} \quad (1)$$

$$\hat{a}_0 = 2\frac{X_m(k_m)}{\sum_{n=0}^{N-1} w(n)} \quad (2)$$

where f_S is the sampling frequency and N is the FFT length.

In Equation (2) we have to scale the FFT magnitude by $w_{\max} = |W(0)|/2$, $W(k)$ being the discrete spectrum of the analysis window $w(n)$, since a sine of amplitude 1 leads to a modulated window with a maximum of $w_{\max}$.

2.2. Parabolic Interpolation

In the magnitude spectrum[2] $|X(e^{j\Omega})|$, the shape of the main lobe of most analysis windows looks like a parabola in the dB scale. For each spectral peak, the parabolic interpolation uses, after the FFT (see above), the main bin and its left and right neighbors in a curve-fitting process with a parabola, using the Brent method [1] to estimate the maximum of the parabola. In order to increase the precision of the parabolic interpolation, zero-padding can be used as in SMS [2, 3].

In the FFT magnitude a search for local maxima is performed. With the FFT magnitude in dB, $X_{\text{dB}}(k) = 20\log_{10}(|X(k)|)$, for each detected maximum (at FFT index k_m) we define

$$A_1 = X_{\text{dB}}(k_m - 1), A_2 = X_{\text{dB}}(k_m), A_3 = X_{\text{dB}}(k_m + 1). \quad (3)$$

Fitting a parabola $f(k)$ through these three points yields the frequency difference in FFT bins

$$d = \frac{1}{2}\frac{A_1 - A_3}{A_1 - 2A_2 + A_3} \quad (4)$$

by detecting the position of the maximum of the parabola. The corrected amplitude is the value of the parabola at its maximum

$$A_{\text{dB}} = f(k_m + d) = A_2 - \frac{d}{4}(A_1 - A_3). \quad (5)$$

[1]In this paper we use the original name of the window designed by the Austrian meteorologist Julius von Hann. It is also often referenced as "Hanning" window.

[2]Considering the discrete-time Fourier transform with the normalized frequency $\Omega = 2\pi f/f_S$.

Then, the difference

$$\Delta A_{\text{dB}} = f(k_m + d) - f(k_m + 1 + d) \tag{6}$$

is compared to the corresponding difference of the reference window spectrum $\Delta A_{\text{ref}} = W_{\text{dB}}(0) - W_{\text{dB}}(1)$. If $|\Delta A_{\text{dB}} - \Delta A_{\text{ref}}|$ is above a certain threshold, the peak is omitted since it is supposed that the FFT maximum is not produced by a sinusoid in the original signal.

Although several windows may be used with this method as for example a truncated Gaussian or a Hann window, we designed a special window. This new window $w(n)$ is shown in Figure 1(a), Figure 1(b) shows its Fourier transform. The main lobe of this window is very close to an exact parabola, Figure 1(c) depicts the error between the original Fourier transform and the polynomial fit using the three main lobe FFT points. An example with an analyzed sinusoid at FFT bin 10.3 is considered in Figure 1(d): the FFT data (circles) an the fitted parabola are shown.

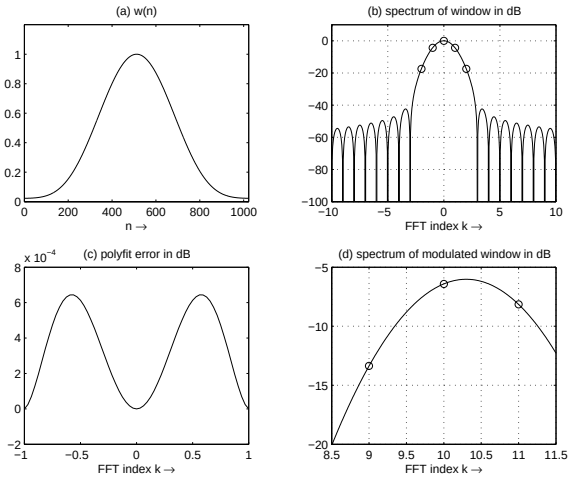

Figure 1: *Window for parabolic interpolation using an FFT length of $N = 1024$. (a) window function $w(n)$ in time domain, (b) discrete-time Fourier transform and FFT values (circles) of $w(n)$, (c) error by fitting a parabola through the FFT data, (d) FFT values (circles) and fitted parabola for a sinusoid at FFT bin $k_0 = 10.3$.*

The performance of this method highly depends on the window function used. Probably a further optimized window can be designed to improve the accuracy of this method.

In our survey, we do not use the amplitude value from Equation (5) for this method, but the amplitude correction as explained in Section 3 is applied. This gives better results in comparison to the original amplitude estimation.

2.3. Triangle Algorithm

The triangle algorithm [4] is named after the shape of the used window function in the frequency domain. This window can be described by two lines in the frequency domain. A multiplication of a sinusoid with the corresponding window function in the time domain results in a modulated (shifted) triangle in the frequency domain.

After applying an FFT to the windowed frame, the FFT magnitude is searched for local maxima. For each detected maximum two triangle lines are fitted through the FFT data in the least squared error sense. At the intersection point of the two lines the amplitude and frequency of the sinusoidal component are estimated.

In this survey we use a window function whose FFT data describe a triangle with a slope length of $S = 2$. Thus, the triangle slope has a length of two FFT bins. We have therefore a quite small main lobe width. Figure 2 shows the used window function in the time and frequency domains and one example for analyzing a sinusoid.

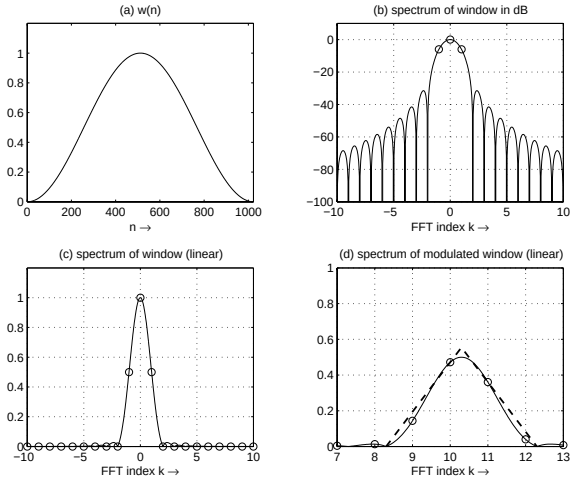

Figure 2: *Window for triangle algorithm using an FFT length of $N = 1024$. (a) window function $w(n)$ in time domain, (b) discrete-time Fourier transform $|W(e^{j\Omega})|$ and FFT values $|W(k)|$ (circles) in dB, (c) $|W(e^{j\Omega})|$ and $|W(k)|$ in linear scale, (d) discrete-time Fourier transform $|X(e^{j\Omega})|$, FFT values $|X(k)|$ (circles), and fitted triangle (dashed) for a sinusoid at FFT bin $k_0 = 10.3$.*

As explained in [4], for each triangle line $S-1$ FFT points are used to reduce the influence of noise. In the special case of $S = 2$, for each line only one FFT point is used. For this reason, these two points are lying ideally on the computed triangle lines.

Since we use another triangle slope length as considered in [4], we use the correction of the amplitude values as explained in Section 3. For the triangle slope length used, this amplitude correction gives better results than the use of a polynomial which is original to the triangle algorithm [4]. For the smaller window main lobe width also the search for local maxima in the FFT magnitude has been simplified compared to the search described in [4].

2.4. Spectral Reassignment

In usual time-frequency representations, the values obtained when decomposing the signal on the time-frequency atoms are assigned to the geometrical center of the cells (center of the analysis window and bins of the Fourier transform). Auger and Flandrin propose in [5] to assign each value to the center of gravity of the cell's energy. The method uses the knowledge of the analytic first derivative $w'(n)$ of the analysis window $w(n)$ in order to adjust the fre-

quency inside the FFT bin. For example, if the analyzed frequency leads to a maximum of magnitude at FFT bin k_m, frequency reassignment is given by the following equation:

$$\hat{f}_0 = k_m \frac{f_S}{N} - \mathrm{Im}\left(\frac{X_{w'}(k_m)}{X_w(k_m)}\right) \cdot \frac{f_S}{2\pi} \tag{7}$$

where $X_w(k)$ and $X_{w'}(k)$ are the FFTs of the signal using $w(n)$ or its first derivative $w'(n)$ as the analysis window, respectively. We will use the Hann analysis window for $w(n)$.

This method is used by Fitz [6] and Peeters [7] for example. Borum and Jensen also present in [8] the use of a similar method for analysis / synthesis.

2.5. Derivative Algorithm

The n-order Fourier analysis [9, 10] shows that it is possible to greatly improve the precision of the classic Fourier analysis by taking advantage of the first n signal derivatives. For $n = 1$, this method is also known as the derivative algorithm. An approximation of the derivative of the input signal $s(n)$ is obtained by

$$s'(n) = f_S[s(n) - s(n-1)]. \tag{8}$$

Both $s(n)$ and $s'(n)$ are windowed and then two FFTs are applied. The accurate frequency of each spectral peak is given by

$$\hat{f}_0 = \frac{f_S}{\pi} \cdot \arcsin\left(\frac{1}{2f_S}\frac{\left|X^1(k_m)\right|}{\left|X^0(k_m)\right|}\right) \tag{9}$$

where $X^n(k)$ denotes the FFT of the n-th signal derivative (so $X^0(k) = X(k)$). The accurate amplitude $\hat{a}_0$ can then be determined from the approximate amplitude together with the deviation $\Delta f = |\hat{f}_0 - k_m f_S/N|$ of the computed frequency $\hat{f}_0$ from the FFT bin frequency

$$\hat{a}_0 = 2\frac{\left|X^0(k_m)\right|}{\left|W\left(e^{j2\pi\Delta f/f_S)}\right)\right|}. \tag{10}$$

This requires the knowledge of the (continuous) power spectrum $W(e^{j\Omega})$ of the analysis window $w(n)$. This method shows excellent results when $w(n)$ is the Hann window.

2.6. FFT with Phase Vocoder Approach

In this method [11, p. 337] the phase information of two FFTs is used to improve the frequency resolution of a plain FFT. For a harmonic signal the fundamental frequency is the derivative of the phase after the time. From the input signal $s(n)$ two frames with a hop size of R samples are taken, and after applying a window to each frame the FFTs of the two frames are computed. The magnitude of the first FFT is searched for local maxima. For each detected maximum k_m the phases φ_1 and φ_2 of the two FFTs are evaluated at this position. With the unwrapped phase φ_{2u} of the second FFT the obtained fundamental frequency is

$$\hat{f}_0 = \frac{f_S}{2\pi} \cdot \frac{\varphi_{2u} - \varphi_1}{R}. \tag{11}$$

In this paper we apply a Hann window before the FFT calculations and we use a hop size of $R = 1$ sample.

3. AMPLITUDE CORRECTION WITH WINDOW MAIN LOBE

In this section we explain in detail the amplitude correction with the window main lobe which is original to the derivative algorithm [9, 10]. Systematical errors of amplitude or frequency depending on the position of the detected frequency between two FFT bins may also be corrected by modeling the error with a polynomial. This error correction is part of the original triangle algorithm [4]. With the used methods we obtain best results with the described amplitude correction using the window main lobe. Thus, this amplitude correction is applied to all used analysis methods except for the plain FFT.

The derivative algorithm corrects the amplitude by using the shape of the window main lobe, see Equation (10). This technique can be applied to other analysis methods in order to improve their precision in amplitude – provided that their precision in frequency is good. More precisely, the spectrum of a windowed, stationary sinusoid is the spectrum $W(e^{j\Omega})$ of the window function $w(n)$, centered about the frequency of the sinusoid, scaled according to the amplitude of the sinusoid, and rotated to the instantaneous phase of the sinusoid at the center of the time-window $w(n)$. Unfortunately, this spectrum $W(e^{j\Omega})$ is also sampled at frequencies corresponding to the bins of the discrete Fourier transform, thus it is evaluated at frequencies $\Omega_k = k\frac{2\pi}{N}$ with the FFT length N and k being the integer-valued FFT-index.

As a consequence, when the frequency of the sinusoid does not exactly correspond to one of the bins of the discrete Fourier transform, the amplitude $|X^0(k_m)|$ measured for the sinusoid at bin k_m is not the amplitude of the sinusoid. In order to determine its exact amplitude, Equation (10) must be used. The spectrum $W(e^{j\Omega})$ of the analysis window $w(n)$ can often be computed analytically. In the case of the periodic Hann window of size N

$$w(n) = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{N}\right)\right] \quad , 0 \le n < N \tag{12}$$

we can express the window by

$$w(n) = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{N}\right)\right] \cdot r(n) \tag{13}$$

with the rectangular window[3]

$$r(n) = \begin{cases} 1 & , 0 \le n \le N \\ 0 & , \text{otherwise.} \end{cases} \tag{14}$$

The discrete-time Fourier transform of $r(n)$ is

$$R(e^{j\Omega}) = \sum_{n=0}^{N} e^{-j\Omega n} = \frac{1 - e^{-j\Omega(N+1)}}{1 - e^{-j\Omega}} \quad , e^{j\Omega} \neq 1 \tag{15}$$

$$= e^{-j\Omega N/2} \cdot \frac{\sin(\Omega\frac{N+1}{2})}{\sin(\frac{\Omega}{2})}. \tag{16}$$

Expressing the window function by

$$w(n) = \frac{1}{2}r(n) - \frac{1}{4}e^{j2\pi n/N}r(n) - \frac{1}{4}e^{-j2\pi n/N}r(n) \tag{17}$$

we get the discrete-time Fourier transform

$$W(e^{j\Omega}) = \frac{1}{2}R(e^{j\Omega}) + \frac{1}{4}R(e^{j(\Omega - \frac{2\pi}{N})}) + \frac{1}{4}R(e^{j(\Omega + \frac{2\pi}{N})}). \tag{18}$$

[3] Although $r(N) = 1$ we get a length-N window with $w(N) = 0$.

Thus, in case of using the Hann window (12), the corrected amplitude is obtained by evaluating Eq. (10) with $W(e^{j\Omega})$ given in Eq. (18).

3.1. Use of a Look-up Table

For more complicated window functions, instead of using an expression of $W(e^{j\Omega})$, a look-up table with the values of $|W(e^{j\Omega})|$ for $0 \leq \Omega < \frac{2\pi}{N}$ (equivalent to the width of one FFT bin) can be computed. For example, using $M = 2^{13} = 8192$ points in this frequency range works well. The table is computed with an FFT of length $M \cdot N$, thus by padding $(M-1) \cdot N$ zeros to $w(n)$; only the first M points of the FFT result are then used for the look-up table. With the values of the look-up table

$$W_t(m) = |W(e^{jm\frac{2\pi}{MN}})| \quad , \; m = 0, \ldots, M-1 \tag{19}$$

and with the estimated frequency difference Δk (in FFT bins, normally $\Delta k < 1$) the amplitude correction is performed by

$$m_0 = \min\{\text{round}(\Delta k \cdot M), M\} \tag{20}$$

$$\hat{a}_0 = 2\frac{|X(k_m)|}{W_t(m_0)}. \tag{21}$$

3.2. Example

Now we consider an example with a sinusoid of amplitude $a_0 = 0.8$ at FFT index $k_0 = 10.3$, $s(n) = a_0 \cos(2\pi k_0 n/N)$, with the FFT length $N = 1024$. After weighting by a Hann window, $x(n) = s(n) \cdot w(n)$, the Fourier transform is

$$X(e^{j\Omega}) = \frac{a_0}{2}W(e^{j(\Omega - 2\pi k_0/N)}) + \frac{a_0}{2}W(e^{j(\Omega + 2\pi k_0/N)}) \tag{22}$$

Figure 3 shows the FFT magnitude $|X(e^{jk2\pi/N})|$ as circles while the dashed line shows the modulated Hann window $\frac{a_0}{2}|W(e^{j(\Omega - 2\pi k_0/N)})|$. Both values are divided by

$$w_{\max} = \frac{1}{2}\sum_{n=0}^{N-1} w(n) \tag{23}$$

which is the maximum of $|X(e^{j\Omega})|$ if using a sinusoid of amplitude 1.

The derivative algorithm detects a maximum at FFT bin $k_m = 10$ and the corrected bin frequency $\hat{k}_0 = 10.2997$ giving a difference of $\Delta k = \hat{k}_0 - k_m = 0.2997$. For the amplitude we have

$$X_{\max} = |X^0(k_m)|/w_{\max} = 0.7546 \tag{24}$$

$$W_{\max} = \frac{1}{2}|W(e^{j2\pi\Delta k/N})|/w_{\max} = 0.9434 \tag{25}$$

$$\hat{a}_0 = \frac{X_{\max}}{W_{\max}} = 0.7999. \tag{26}$$

Thus, instead of taking the amplitude value at the FFT maximum, we recover almost perfectly the original amplitude value by using the estimated frequency of the sinusoid. This amplitude correction method requires therefore a good frequency estimation.

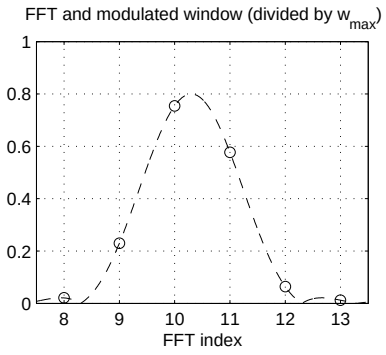

Figure 3: *Spectra of sinusoid at FFT index $k_0 = 10.3$ and use of the Hann window. FFT data (circles) and modulated Hann window (dashed line).*

4. SIMULATION RESULTS

In this section we present the results of different tests. In Section 4.1 first the used measurements are explained while the following subsections describe the different tests.

In the first test (Section 4.2) we use stationary single sinusoids that do not change their frequency during one analysis frame. We use also single sinusoids with additive white noise. As a second test (Section 4.3) we use two stationary sinusoids with varying frequency difference to evaluate the frequency resolution of the different analysis methods. As a third test (Section 4.4) we use non-stationary signals such as sine sweeps, tremolo and vibrato sinusoids whose amplitude and/or frequency change continuously over time. Finally (Section 4.5) we use more complex signals with a high number of harmonics. All tests use a sampling frequency of $f_S = 44.1$ kHz and an FFT length of $N = 1024$.

4.1. Measurements

4.1.1. Frequency and Amplitude Error

For the frequency, we consider the error produced by the method in number of halftones [11, p. 349]. Considering a sine at FFT bin k_0 (as fractional number) resulting in the absolute frequency error Δk_0, the frequency error factor is

$$\gamma_0 = \frac{k_0 + \Delta k_0}{k_0} > 1. \tag{27}$$

With the halftone factor $\gamma_{\text{ht}} = \sqrt[12]{2}$ we get the error in percent of number of halftones as

$$\Delta f_{\text{ht}} = 100\frac{\log(\gamma_0)}{\log(\gamma_{\text{ht}})}. \tag{28}$$

Please notice that humans can recognize a frequency error of 6% of a halftone if they are trained in listening (see [12] for information on the human auditory system).

For the amplitude we use the error in dB

$$\Delta a_{\text{dB}} = \left| 20\log_{10}\left(\frac{\hat{a}_0}{a_0}\right)\right| \tag{29}$$

where a_0 and $\hat{a}_0$ are the reference and estimated amplitudes, respectively.

For both frequency and amplitude error measurements we take the mean value μ, the standard deviation σ and the maximum value over all frames of the simulation.

4.1.2. Decision of Detected Peaks

For reference signals containing more than one sinusoid it is of interest to measure if all sinusoids are detected by the different methods. For this task, we added some post-processing after the extraction of spectral peaks. To decide whether a detected peak corresponds to a sine in the reference signal or not, the following procedure is used for all reference sines of one analysis frame:

- Use only peaks whose frequencies vary at maximum by one FFT bin width from the reference frequency.
- Among the remaining candidates, use only peaks whose amplitudes vary at maximum by 3 dB from the reference amplitude.
- Among the remaining candidates, choose that one with the smallest amplitude and frequency error compared to the reference sinusoid.

Now the last candidate is supposed to be the extracted sinusoid. The peaks omitted by this procedure are classified as "misdetected sines". Table 2 presents some measurements regarding the number of detected spectral peaks per analysis frame.

$\overline{N}_{\text{peaks}}$	Mean number of detected spectral peaks per frame.
$\overline{N}_{\text{missed}}$	Mean number of spectral peaks per frame that are in the reference signal, but not detected by the algorithm.
$\overline{N}_{\text{miss}}$	Mean number of misdetected spectral peaks per frame. Thus, the mean number of spectral peaks per frame that are detected by the algorithm, but not present in the reference signal.

Table 2: *Measurements for the detection of spectral peaks.*

Notice that for frequency and amplitude errors only those sines are used which correspond to the reference sines according to the above described procedure.

4.2. Stationary Single Sinusoids

In our first test, we use sines with constant amplitude and frequency during one frame. We generated a number of $L = 2090$ sines with frequencies f_0 between 215 and 4321 Hz, and with a sine amplitude of $a_0 = 1$. With the frequency in FFT bins $k_0 = f_0/f_S \cdot N$ we get the position between two FFT bins as $k_{\text{frac}} = k_0 - \lfloor k_0 \rfloor$. Fig. 4 shows the probability functions of f_0 and k_{frac}, respectively. Thus, we emphasize the low frequencies according to their appearance in natural sounds.

We use pure sines and sines with additive Gaussian white noise with an SNR $= 12$ dB. The results for the frequency and amplitude errors are shown in Tables 3 and 4. When using pure sines, each method detects exactly one sine in all frames. For the sines with added noise, Table 5 shows the mean number of detected sinusoids in each frame for the different methods. The maximum amplitude of misdetected sines is in the range of -26 to -22 dB

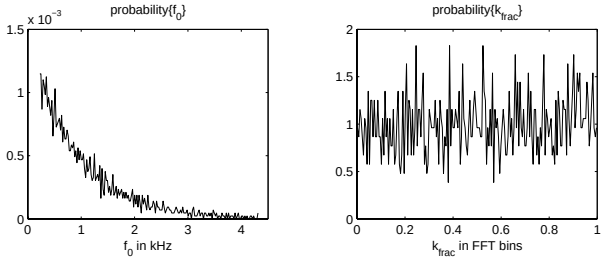

Figure 4: *Probabilities of reference frequencies and of positions between two FFT bins.*

	pure sines			SNR = 12 dB		
No.	μ	σ	max	μ	σ	max
1	28.11	28.12	149.76	28.12	28.14	151.69
2	0.054	0.118	1.096	0.851	1.037	9.725
3	0.006	0.016	0.136	0.725	0.849	7.798
4	0.048	0.115	1.097	0.623	0.768	8.061
5	0.048	0.115	1.100	0.624	0.769	8.045
6	0.048	0.115	1.099	0.623	0.768	8.046

Table 3: *Frequency error in percentage of halftones for pure sines and for sines with white noise at SNR = 12 dB.*

	pure sines			SNR = 12 dB		
No.	μ	σ	max	μ	σ	max
1	0.470	0.428	1.424	0.482	0.420	1.545
2	0.001	0.001	0.008	0.076	0.056	0.334
3	0.000	0.000	0.001	0.072	0.054	0.321
4	0.001	0.002	0.017	0.067	0.050	0.292
5	0.001	0.002	0.017	0.067	0.050	0.293
6	0.001	0.002	0.017	0.067	0.050	0.292

Table 4: *Amplitude error in dB for pure sines and for sines with white noise at SNR = 12 dB.*

No.	1	2	3	4	5	6
$\overline{N}_{\text{peaks}}$	140.1	40.8	52.2	139.4	118.9	134.9

Table 5: *Mean number of detected sines per frame for sines at SNR = 12 dB.*

for all methods. From these results the advantage in frequency and amplitude precision of methods 2–6 over the plain FFT is obvious.

For pure sines, method 3 (triangle algorithm) produces the best results for both frequency and amplitude accuracy. In the noisy case, methods 4–6 have slightly better results, although the maximum frequency error is also a little better with method 3. Methods 2 and 3 have the advantage that the number of misdetected sinusoids is much lower compared to the other methods, see Table 5.

This test shows very similar results for methods 4–6, which all require two FFTs and use the phase information in some way.

4.3. Two Stationary Sinusoids with Varying Frequency Difference

In this test we use in each analysis frame two sinusoids. One sinusoid is at FFT bin $k_{0,1} = 10.3$ (444 Hz) while the second sine is at $k_{0,2}$ changing from 5 to 16 (215–689 Hz). We use only frames where the two sinusoids have a frequency difference greater than one percent of a halftone (to omit the superposition of two sines with the same frequency). The test signal has a length of $L = 1099$ analysis frames.

Table 6 presents the results for two sinusoids which both have an amplitude of 1, and Figure 5 shows the estimated spectral peaks for all methods of the performed test.

No.	1	2	3	4	5	6
$\overline{N}_{\text{peaks}}$	1.72	1.29	1.40	1.71	1.71	1.68
$\overline{N}_{\text{missed}}$	0.36	0.79	0.71	0.35	0.35	0.38
$\overline{N}_{\text{miss}}$	0.17	0.13	0.14	0.15	0.15	0.15
$\overline{\Delta f_{\text{ht}}}$	51.43	6.79	8.92	12.98	12.95	12.49
$\Delta f_{\text{ht,max}}$	178.6	73.0	188.2	175.7	178.6	176.0
$\overline{\Delta a_{\text{dB}}}$	0.68	0.21	0.17	0.30	0.31	0.29

Table 6: *Results for two sines with varying frequency difference.*

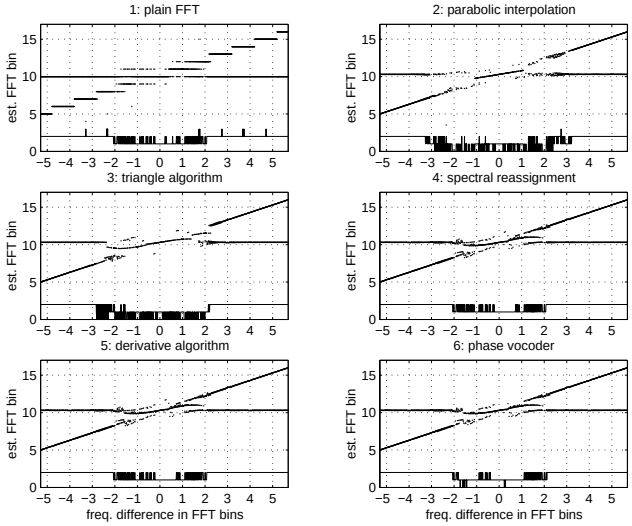

Figure 5: *Estimated spectral peaks over the frequency difference of two sinusoids. The bottom graph in each plot indicates the number of detected peaks.*

From the presented results it is easy to see that methods 1 and 4–6 have a frequency resolution of appr. two FFT bins, while method 2 needs a distance of more than 3 FFT bins, and method 3 has a frequency resolution of appr. 3 FFT bins. The frequency resolution is also reflected by the results of $\overline{N}_{\text{peaks}}$ and $\overline{N}_{\text{missed}}$ in Table 6. In that table, the mean frequency and amplitude errors for methods 2 and 3 are better than for the others, but for the error measurements only the detected sinusoids are used. Thus, altogether methods 4–6 give the best results for this test.

Please notice that the frequency resolution depends also on the amplitudes of the used sines.

4.4. Non-Stationary Single Sinusoids

In this test we use a single sine signal whose amplitude and/or frequency is changing continuously over time. We consider four test signals: sweep, tremolo, vibrato, and combination of tremolo / vibrato. For all test signals we use a signal duration of $\tau_s = 2$ seconds.

In general we have the amplitude $a(t)$ over time and the frequency $f(t)$ over time leading to the continuous-time signal

$$s(t) = a(t)\cos(2\pi\varphi(t)) \tag{30}$$

whose phase $\varphi(t)$ has to satisfy

$$f(t) = \frac{\mathrm{d}\varphi(t)}{\mathrm{d}t}, \tag{31}$$

thus the frequency is the derivative of the phase.

For the sine sweep we have a constant amplitude $a(t) = a_0$ and the frequency increases linearly from f_0 to f_1 which leads to

$$f(t) = f_0 + \frac{f_1 - f_0}{\tau_s} \cdot t \tag{32}$$

$$s(t) = a_0 \cos\left(2\pi\left[f_0 t + \frac{f_1 - f_0}{2\tau_s} t^2\right]\right). \tag{33}$$

We use $a_0 = 0.8$, $f_0 = 440$ Hz and $f_1 = 880$ Hz.

For a tremolo sound the amplitude is oscillating and the sine frequency $f(t) = f_0$ is constant which results in

$$a(t) = a_0 + a_1 \sin(2\pi f_t t) \tag{34}$$

$$s(t) = a(t)\cos(2\pi f_0 t)\,. \tag{35}$$

The oscillating amplitude a_1 is also called tremolo depth, and its frequency f_t is the tremolo frequency. We use $a_0 = 0.8$, $f_0 = 440$ Hz, $a_1 = 0.15$ and $f_t = 5$ Hz.

In a vibrato sound the amplitude $a(t) = a_0$ is constant, and the sine frequency $f(t)$ is oscillating which gives

$$f(t) = f_0 + f_1 \cos(2\pi f_v t) \tag{36}$$

$$s(t) = a_0 \cos\left(2\pi\left[f_0 t + \frac{f_1}{2\pi f_v}\sin(2\pi f_v t)\right]\right). \tag{37}$$

Here f_1 is the vibrato depth, and f_v is its frequency. We use $a_0 = 0.8$, $f_0 = 440$ Hz, $f_1 = 10$ Hz and $f_v = 10$ Hz.

For the combined tremolo / vibrato sound, both amplitude and frequency are oscillating according to Equations (34) and (36) which leads to

$$s(t) = a(t)\cos\left(2\pi\left[f_0 t + \frac{f_1}{2\pi f_v}\sin(2\pi f_v t)\right]\right). \tag{38}$$

For this test signal we combine the parameters given above, thus $a_0 = 0.8$, $f_0 = 440$ Hz, $a_1 = 0.15$, $f_t = 5$ Hz, $f_1 = 10$ Hz, and $f_v = 10$ Hz.

For the analysis we use a hop size of 64 samples, thus the analyzing frames overlap by $1024 - 64 = 960$ samples. This leads to a number of $L = 1363$ analysis frames. As reference amplitude and frequency values their mean values during one analysis frame are used. Tables 7, 8, 9, 10 show the results of amplitude and frequency error for the four test signals.

	freq. err. in % of halftones			ampl. error in dB		
No.	μ	σ	max	μ	σ	max
1	29.31	18.32	80.30	0.474	0.421	1.423
2	0.043	0.050	0.325	0.001	0.001	0.005
3	0.010	0.007	0.035	0.001	0.000	0.001
4	0.025	0.027	0.176	0.001	0.001	0.007
5	0.031	0.028	0.195	0.001	0.001	0.005
6	0.028	0.027	0.183	0.001	0.001	0.007

Table 7: *Frequency error in percentage of halftones and amplitude error in dB for a sine sweep.*

	freq. err. in % of halftones			ampl. error in dB		
No.	μ	σ	max	μ	σ	max
1	36.35	0.00	36.35	0.266	0.016	0.292
2	0.046	0.022	0.074	0.017	0.009	0.036
3	0.078	0.038	0.153	0.016	0.008	0.030
4	0.088	0.060	0.234	0.014	0.007	0.026
5	0.114	0.071	0.273	0.014	0.007	0.026
6	0.088	0.060	0.235	0.014	0.007	0.026

Table 8: *Frequency error in percentage of halftones and amplitude error in dB for a tremolo sound.*

	freq. err. in % of halftones			ampl. error in dB		
No.	μ	σ	max	μ	σ	max
1	35.47	24.29	69.78	0.407	0.402	1.113
2	1.654	0.809	2.745	0.003	0.003	0.010
3	1.541	0.771	2.615	0.003	0.002	0.006
4	1.336	0.674	2.384	0.005	0.003	0.012
5	1.339	0.676	2.396	0.005	0.003	0.013
6	1.336	0.674	2.385	0.005	0.003	0.012

Table 9: *Frequency error in percentage of halftones and amplitude error in dB for a vibrato sound.*

	freq. err. in % of halftones			ampl. error in dB		
No.	μ	σ	max	μ	σ	max
1	35.47	24.29	69.78	0.412	0.397	1.113
2	1.656	0.810	2.746	0.017	0.010	0.034
3	1.549	0.777	2.795	0.015	0.009	0.029
4	1.332	0.671	2.365	0.014	0.010	0.027
5	1.339	0.675	2.464	0.014	0.009	0.028
6	1.333	0.671	2.369	0.014	0.010	0.027

Table 10: *Frequency error in percentage of halftones and amplitude error in dB for a sound with both tremolo and vibrato.*

For the sine sweep, method 3 (triangle algorithm) gives the best results for amplitude and frequency evaluation. The other methods (except the plain FFT) give only slightly higher errors.

In case of the tremolo sound, method 2 (parabolic interpolation) produces the smallest frequency error although the other methods (except plain FFT) produce only slightly higher errors. The amplitude error is almost the same for all methods (except plain FFT).

For the vibrato sound, again all methods (except plain FFT) produce quite similar results. This time, methods 4–6 are better than the other methods for the frequency error. As expected, the produced frequency error is higher as for the tremolo sound, while the amplitude error is smaller for all methods.

If using both tremolo and vibrato, the results of the previous simulation are almost exactly recovered. Thus we have the frequency error of the vibrato sound and the amplitude error of the tremolo sound. Methods 4–6 show almost the same results and are favored a little bit over the other methods.

4.5. Harmonic Natural-Like Sounds

In our last test we use signals which are very close to natural signals. In order to evaluate the different methods we need reference values for the amplitudes and frequencies of the sinusoids contained in the signals. For this reasons it is not possible to use recorded natural sounds. For our survey we use sounds which are synthesized from amplitude and frequency values that change every 64 samples. We use also a hop size of 64 samples for analyzing these sounds. The used sounds are:

- human voice: $L = 989$ frames, 40.06 sines per frame
- saxophone: $L = 1947$ frames, 30.46 sines per frame
- guitar: $L = 4722$ frames, 11.75 sines per frame

Tables 11, 12, 13 show the results containing the measurements introduced in Table 2.

No.	1	2	3	4	5	6
$\overline{N}_{\text{peaks}}$	38.4	33.3	30.5	38.4	38.4	38.3
$\overline{N}_{\text{missed}}$	8.30	10.16	11.68	7.79	7.80	7.85
$\overline{N}_{\text{miss}}$	6.63	3.37	2.07	6.11	6.11	6.06
$\overline{\Delta f_{\text{ht}}}$	12.51	4.42	4.46	4.59	4.59	4.59
$\Delta f_{\text{ht,max}}$	209.5	65.8	68.5	73.2	74.5	73.4
$\overline{\Delta a_{\text{dB}}}$	0.86	0.75	0.71	0.77	0.77	0.77

Table 11: *Results for natural-like sound of human voice.*

No.	1	2	3	4	5	6
$\overline{N}_{\text{peaks}}$	29.9	27.6	24.7	29.9	29.9	29.9
$\overline{N}_{\text{missed}}$	2.49	4.13	6.46	2.20	2.22	2.20
$\overline{N}_{\text{miss}}$	1.89	1.23	0.71	1.60	1.62	1.60
$\overline{\Delta f_{\text{ht}}}$	7.48	1.27	1.27	1.26	1.37	1.26
$\Delta f_{\text{ht,max}}$	86.7	28.3	4.8	29.4	29.8	29.5
$\overline{\Delta a_{\text{dB}}}$	0.92	0.72	0.68	0.73	0.72	0.73

Table 12: *Results for natural-like saxophone sound.*

For all methods (except plain FFT) the frequency and amplitude accuracies do not differ very much. But the number of missed peaks differs for the methods due to their frequency resolution. Regarding the number of detected sines, methods 4 to 6 produce the best results (lowest number of $\overline{N}_{\text{missed}}$). Please notice again, that the amplitude and frequency errors are only calculated for those peaks which correspond to sines in the reference signal. For this reason, methods 2 and 3 sometimes give better results for frequency and amplitude, but they do not detect as much sines as the

No.	1	2	3	4	5	6
$\overline{N}_{\text{peaks}}$	11.7	11.0	9.8	11.7	11.7	11.7
$\overline{N}_{\text{missed}}$	0.10	0.76	2.00	0.09	0.09	0.10
$\overline{N}_{\text{miss}}$	0.06	0.03	0.03	0.06	0.06	0.06
$\overline{\Delta f_{\text{ht}}}$	27.94	0.83	0.76	1.72	1.73	1.70
$\Delta f_{\text{ht,max}}$	193.2	31.7	15.5	42.3	42.4	42.1
$\overline{\Delta a_{\text{dB}}}$	0.66	0.21	0.19	0.23	0.23	0.23

Table 13: *Results for natural-like guitar sound.*

other methods. But also the number of misdetected peaks is higher with methods 4 to 6.

The amplitude values of the missed or misdetected peaks are quite similar for all methods. Table 14 shows a summary of the amplitude values in dB for the three considered signals. It seems that for the used signals most of the missed peaks have a very low amplitude, and the methods use a threshold for omitting any signal content below this threshold (normally -80 dB). Only few of the missed spectral peaks have a quite high amplitude. These peaks are not detected due to the frequency resolution of the different methods.

	missed		misdetected (miss)	
signal	mean	max	mean	max
voice	-71…-68	-13	-61…-54	-16
saxophone	-75…-70	-17.5	-63…-57	-14
guitar	-68…-64	-22	-64…-51	-19

Table 14: *Amplitude values in dB of missed and misdetected peaks for all methods.*

5. CONCLUSION

Spectral analysis and more precisely sine extraction is a key point in many applications such as spectral modeling, pitch tracking or digital audio effects in general.

In this paper we compared six of the most used spectral analysis methods. All of these methods – based on the FFT – are very fast, but their precisions in frequency and amplitude are not equivalent. This study can be used as a reference for anyone willing to implement such a method, thus facilitating the choice of the method which is the most adapted to one's specific need.

As summary of our simulations, the methods based on the phase information of the FFT, namely the spectral reassignment, the derivative algorithm, and the phase vocoder approach (method numbers 4, 5 and 6 in our survey) give the best results regarding frequency resolution while having very small frequency and amplitude errors. It is interesting to notice that these three methods give almost the same results in the performed tests. It is subject of our current research to show that these three methods may be equivalent even from their theoretical foundations, at least when the Hann analysis window is used.

6. REFERENCES

[1] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, "Minimization or maximization of functions," in *Numerical Recipes in C (The Art of Scientific Computing)*, chapter 10, pp. 402–405. Cambridge University Press, USA, 2nd edition, 1992.

[2] Xavier Serra, *A System for Sound Analysis/ Transformation/ Synthesis Based on a Deterministic plus Stochastic Decomposition*, Ph.D. thesis, CCRMA, Department of Music, Stanford University, 1989.

[3] Xavier Serra, "Musical sound modeling with sinusoids plus noise," in *Musical Signal Processing*, C. Roads, S.T. Pope, A. Piccialli, and G. De Poli, Eds., chapter 3, pp. 91–122. Swets & Zeitlinger, Lisse, the Netherlands, 1997.

[4] F. Keiler and U. Zölzer, "Extracting sinusoids from harmonic signals," *Journal of New Music Research, Special Issue: "Musical Applications of Digital Signal Processing"*, vol. 30, no. 3, pp. 243–258, Sept. 2001.

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

[6] Kelly R. Fitz, *The Reassigned Bandwith-Enhanced Method of Additive Synthesis*, Ph.D. thesis, University of Illinois, 1999.

[7] Geoffroy Peeters and Xavier Rodet, "SINOLA: A New Analysis/ Synthesis Method using Spectrum Peak Shape Distortion, Phase and Reassigned Spectrum," in *Proc. ICMC*, Beijing, China, October 1999, ICMA, pp. 153–156.

[8] Stefan Borum and Kristoffer Jensen, "Additive Analysis/ Synthesis Using Analytically Derived Windows," in *Proc. DAFx-99*, Trondheim, Norway, December 1999, Norwegian University of Science and Technology (NTNU) and COST, pp. 125–128.

[9] Sylvain Marchand, "Improving Spectral Analysis Precision with an Enhanced Phase Vocoder Using Signal Derivatives," in *Proc. DAFx-98*, Barcelona, Spain, Nov. 1998, Audiovisual Institute, Pompeu Fabra University and COST, pp. 114–118.

[10] Myriam Desainte-Catherine and Sylvain Marchand, "High Precision Fourier Analysis of Sounds Using Signal Derivatives," *JAES*, vol. 48, no. 7/8, pp. 654–667, July/August 2000.

[11] D. Arfib, F. Keiler, and U. Zölzer, "Source-filter processing," in *DAFX – Digital Audio Effects*, U. Zölzer, Ed., chapter 9, pp. 299–372. J. Wiley & Sons, Chichester, 2002.

[12] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer Verlag, 1990.