Assessing the Quality of the Extraction and Tracking of Sinusoidal Components: Towards an Evaluation Methodology

Mathieu Lagrange, Sylvain Marchand

ASSESSING THE QUALITY OF THE EXTRACTION AND TRACKING OF SINUSOIDAL COMPONENTS: TOWARDS AN EVALUATION METHODOLOGY

Mathieu Lagrange, Sylvain Marchand

SCRIME – LaBRI, University of Bordeaux 1
351 cours de la Libération, F-33405 Talence cedex, France
firstname.name@labri.fr

ABSTRACT

In this paper, we introduce two original evaluation methods in the context of sinusoidal modeling. The first one assesses the quality of the extraction of sinusoidal components from short-time signals, whereas the second one focuses on the quality of the tracking of these sinusoidal components over time.

Each proposed method intends to use a unique cost function that globally reflects the performance of the tested algorithm in a realistic framework. Clearly defined evaluation protocols are then proposed with several test cases to evaluate most of the desired properties of extractors or trackers of sinusoidal components.

This paper is a first proposal to be used as a starting point in a sinusoidal analysis / synthesis contest to be held at DAFx'07.

1. INTRODUCTION

The sinusoidal model is widely known, and its usability and versatility do not have to be proved. Many tools [1, 2, 3, 4] have been proposed to analyze and synthesize speech and musical sounds using this model for various applications from sound modifications [5, 6] to audio coding [7, 8].

The analysis / synthesis chain is generally divided into three main stages, as shown in Figure 1. Despite the numerous approaches that have been proposed in the literature to implement each of the three stages, the issue of evaluating and comparing the performance of these proposals is still an open question, at least for the first two stages. However, we think that there is a strong need to be able to assess the performance of each of these two stages in case of general audio processing uses.

Ideally, one would like general purpose evaluation methods with the following properties:

1. each part of the analysis chain is evaluated without the influence of the others;
2. the test cases should be realistic: the number of sinusoids is unknown and their parameters can be modulated.

In an effort to address these issues, we propose in this paper two original methods: the first one assesses the quality of sinusoidal components extractors, whereas the second one assesses the quality of trackers of sinusoidal components.

The sinusoidal model and the complete analysis / synthesis chain is presented in Section 2 where the output of each stage is formalized in terms of the set theory. Existing methodologies that evaluate the performance of sinusoidal components extractors are then reviewed and discussed in Section 3. We then show that the issue of quality measurement can be considered as a set comparison problem. From these remarks, an original evaluation metric is proposed. This metric is then considered in various test cases that build a complete evaluation framework of sinusoidal components extractors. A similar approach (review of existing methodologies, new evaluation method, and associated protocol) is done in Section 4 for the trackers of these sinusoidal components.

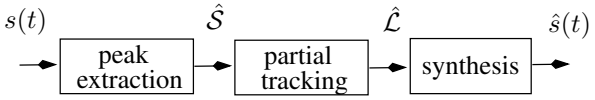


Figure 1: *The sinusoidal analysis / synthesis chain. The peak extractor extracts sinusoidal components from short-time signals. These components belong to the $\mathcal{S}$ set. Components of successive frames are then considered by some tracking algorithm to estimate the evolutions of the parameters of sinusoidal components (the $\mathcal{L}$ set) over time.*

2. SINUSOIDAL MODELING

Additive synthesis is the original spectrum modeling technique. It is rooted in Fourier's theorem, which states that any periodic function can be modeled as a sum of sinusoids at various amplitudes and harmonic frequencies. For stationary pseudo-periodic sounds, these amplitudes and frequencies continuously evolve slowly with time, controlling a set of pseudo-sinusoidal oscillators commonly called *partials*. The audio signal s can be calculated from the additive parameters using Equations 1 and 2, where N is the number of partials and the functions f_n, a_n, and ϕ_n are the instantaneous frequency, amplitude, and phase of the n-th partial, respectively. The N pairs (f_n, a_n) are the parameters of the additive model and represent points in the frequency-amplitude plane at time t. This representation is used in many analysis / synthesis programs such as AudioSculpt [1], Lemur [2], SMS [3], or InSpect [4].

$$s(t) = \sum_{n=1}^{N} a_n(t)\ \cos(\phi_n(t)) \quad (1)$$

$$\phi_n(t) = \phi_n(0) + 2\pi \int_0^t f_n(u)\ du \quad (2)$$

As presented in the introduction, the sinusoidal analysis / synthesis chain is generally divided into three main stages, as shown in Figure 1.

2.1. The Extraction of Peaks

The first stage, called "peak extraction", is intended to determine the number of sinusoids from a short-time signal and to estimate the parameters of each sinusoid within the interval of observation. The term *peak* is due to the fact that the power spectrum of a sinusoid mainly consists of a prominent peak (local maximum).

At each frame, some sinusoidal components are detected and their parameters are estimated. In this paper, a peak p_i^k (peak number i of frame number k) is defined by four parameters: the frequency f_i^k (considered as constant during the observation interval), amplitude a_i^k, phase ϕ_i^k, and confidence c_i^k:

$$p_i^k = \left(f_i^k, a_i^k, \phi_i^k, c_i^k\right) \tag{3}$$

where f_i^k, a_i^k, and c_i^k are normalized parameters between 0 and 1. The c_i^k parameter denotes the degree of confidence given by the extraction algorithm. A small c_i^k indicates that the peak p_i^k should not be trusted (e.g. may belong to some noisy component).

The peaks of the same frame may be clustered in a frame set and the resulting sets may in turn be grouped to build the (multi) set of peaks $\mathcal{S}$:

$$\mathcal{S} = \bigcup_k \mathcal{S}_k \quad \text{where} \quad \mathcal{S}_k = \bigcup_i \left\{p_i^k\right\} \tag{4}$$

A review of the methods that permit such extraction from audio signals is out of the scope of this paper, see [9, 10] for reviews and references about methods based on the Fourier spectrum and see for example [11] concerning high-resolution estimation methods. In this paper, we aim at finding a way of *evaluating* such methods.

2.2. The Tracking of Partials

The second stage is known as "partial tracking". This part identifies continuities between peaks of consecutive frames and therefore determines the continuous evolutions of sinusoids over time. The term *partial* is due to the fact that many partials are usually needed at the same time to model one musical or voice tone.

Formally, a partial is a vector of peaks (of confidence 1) with successive time indices:

$$P_n(m) = \{F_k(m), A_k(m), \Phi_k(m)\},\ \forall m \in [b_n, \cdots, b_n + l_n - 1] \tag{5}$$

where P_n is the partial number n, of length l_n, and that appeared (was born) at frame index b_n; $P_n(m)$ is the peak of time index m of this partial. Following these notations, a peak that does not belong to any partial is noted p and noted P otherwise. For example, $p_i^m = P_n(m)$ means that the i-th peak of frame m has been assigned to the partial number n.
The set of partials that represents the entire sound is defined as:

$$\mathcal{L} = \bigcup_{n=1}^{N} P_n \tag{6}$$

2.3. Synthesis

The last stage synthesizes back the output audio signal using the estimated sinusoidal parameters. The continuous evolutions of the sinusoidal parameters of Equations 1 and 2 are computed from the parameters of the partials using interpolation methods [12, 13] and efficient synthesizers are then used to obtain the output signal $\hat{s}(t)$.

The performance of different methods [14, 15] that implement this last stage can be evaluated and compared using some clearly defined protocols. To do so, the synthesized sound $\hat{s}(t)$ is compared to the original one $s(t)$ using some objective or subjective criteria.

The Reconstruction Signal-to-Noise Ratio (R-SNR) can be considered as an objective criterion:

$$\text{R-SNR} = 10 \log 10 \left(\sum_{l=0}^{L-1} (s(l) - \hat{s}(l))^2 \Big/ \sum_{l=0}^{L-1} s^2(l) \right) \tag{7}$$

where s is the original (short-time) signal and $\hat{s}$ the synthesized one, both signals consisting of L samples. This objective criterion can be easily computed but has the disadvantage of being sometimes perceptively irrelevant. Alternatively, some expert listeners may be asked to judge the quality of $\hat{s}$ versus s using software tools such as [16].

Approximately the same protocol can be considered to compare the performance of the other stages of the chain. If we want to compare several extractors (*resp.* trackers), a given tracker (*resp.* extractor) and a synthesis module are chosen and a switch is done between the several peak extractors (*resp.* partial trackers). However, the influence of the chosen tracker (*resp.* extractor) and synthesis modules may favor a specific extractor (*resp.* tracker) and these biases cannot be quantified easily.

Alternatively, we introduce in this paper flexible evaluation methods that do not compare time-signals and thus do not need implementations of the other stages to compare a specific stage of the chain.

3. PEAK EXTRACTION

Extractors are generally designed to detect the right number of sinusoids and to estimate the parameters of each sinusoid precisely.

3.1. Existing Evaluation Methods

Therefore, existing evaluation methods usually decorrelate the parameter estimation issue from the detection one and evaluate them independently.

3.1.1. Frequency Estimation

As far as the estimation of parameters is concerned, we generally focus on the frequency one since the others (amplitude and phase) can be found by Maximum Likelihood (ML) methods once the frequency is estimated.

Let us consider a (real) sinusoid x (of amplitude 1) in a Gaussian noise y (of variance σ^2), both (short-time) signals consisting of L samples:

$$x(l) = \sin(j\omega l + \Phi) \tag{8}$$

$$y(l) = 10^{-\text{SNR}/20}/\sqrt{2} \cdot z(l) \tag{9}$$

where ω is the frequency (in radians per sample) and z is a Gaussian noise of variance 1. The variance of the signal part x is $1/2$, and the variance of the noise part y is $\text{var}(y) = \sigma^2 = 10^{-\text{D-SNR}/10}/2$. The analyzed signal is $s = x + y$.

For the case of the estimation of the frequency ω of a sinusoid in noise, the lower Cramér-Rao Bound (CRB) [17] is:

$$\mathrm{var}(\hat{\omega}) \geq \frac{24\,\sigma^2}{a^2 L(L^2-1)} = \frac{12}{L(L^2-1)}\,10^{-\text{D-SNR}/10} \qquad (10)$$

where a is the amplitude of the sinusoid (here $a = 1$), and the Degradation-SNR is given by the following equation:

$$\text{D-SNR} = 10\log 10\left(\sum_{l=0}^{L-1} y^2(l) \Big/ \sum_{l=0}^{L-1} x^2(l)\right) \qquad (11)$$

We can easily show that, in the log scales, the CRB in function of the SNR is a line, see Figure 2.

Therefore, the variance of the error versus the SNR gives an indication of the precision of the tested estimator, *i.e.* the closer to the CRB, the better. This framework has also been used to evaluate the frequency resolution of estimators in [10], by considering two sinusoids instead of only one. However, such tests are often done using complex exponentials instead of real sinusoids.

The clear mathematical formulation of this evaluation method is convenient for mathematical derivations of theoretical performance estimation bounds. However, this unrealistic test case has the major disadvantage of being over-simplified and may therefore lead to over-specialized estimators whose applicability to musical and speech signal analysis remains to be proved.

3.1.2. Detection

Keiler *et al.* use in [9] an algorithm similar to the one proposed in this paper, where extracted peaks are compared to the original ones and the following quantities are computed: the mean number of detected peaks per frame, the mean number of peaks per frame that are in the reference signal but not detected by the algorithm, and the mean number of peaks that are detected by the algorithm but not present in the reference signal.

Hainsworth in [10] also uses two quantities: the number of correctly identified sinusoids and the number of falsely detected sinusoids. However, this "correctness" is not clearly defined.

3.2. Proposed Evaluation Methodology

As remarked by Hainsworth in [10], even if the problems of parameter estimation and component detection can be separated, they are in practice inextricably linked. We therefore propose in this section a method to evaluate globally the performance of a peak extractor without considering the other stages of the analysis / synthesis chain.

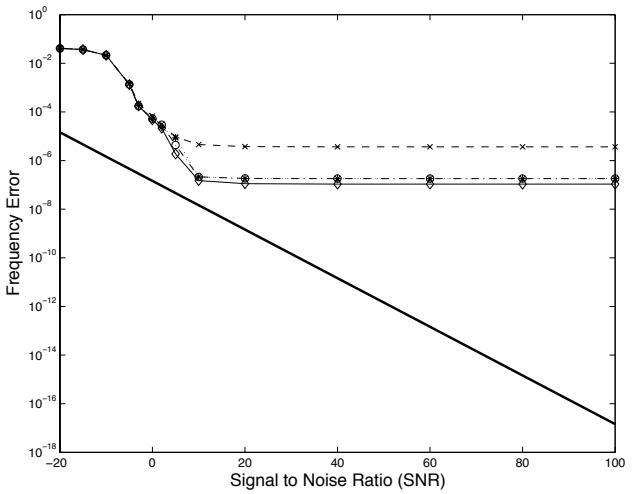
Given a set of peaks $\mathcal{S}_k$ of cardinal $\#\mathcal{S}_k = N_k$, the original (short-time) sound is synthesized and possibly degraded. The tested extractor is then used to extract another set of sinusoids $\hat{\mathcal{S}}_k$ with cardinal M_k. The frame index k will be removed in the following for the sake of clarity. The closer this set is to the original, the better the extractor is.

The issue here is then to be able to compare these two sets. Following the remarks of Section 3.1, we consider only the frequency and confidence parameters for the evaluation.

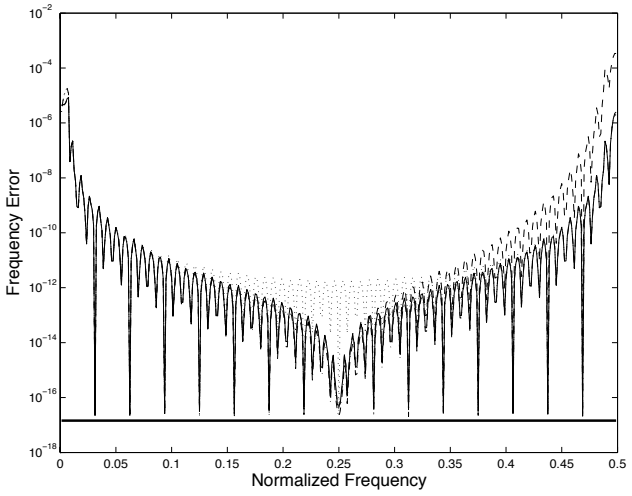
3.2.1. Unitary Case

Let us consider a simple case, where only one peak is in the original set and the estimated one: $N = M = 1$. To compute the imprecision cost between a peak of the original set p_i and a peak in the estimated set $\hat{p}_j$, we use only the frequency parameter. Since we consider only real signals, the normalized frequencies are between 0 and 0.5. A normalization factor of 2 is then used to obtain a cost function between 0 and 1. More precisely:

$$c(p_i, \hat{p}_j) = (2(f_i - \hat{f}_j))^2 \qquad (12)$$



(a)



(b)

Figure 2: *(a) Performance comparison of several estimators from [18] for the analysis of a real sinusoid with frequency lying in the whole (normalized) frequency range* $]0, 0.5[$*: the reassignment method (dotted line with *), the phase-vocoder estimator (dash-dotted line with o), the derivative estimator (dashed line with x), and the trigonometric estimator (solid line with* $\diamond$*). The CRB is plotted with a double solid line. (b) Performance of the tested estimators at SNR=*100 *dB versus the frequency of the analyzed sinusoid (symbols are not plotted here for the sake of clarity).*

3.2.2. General Case

Let us now consider the general case where N is set arbitrarily. Since all the sinusoids of the original set exist, they have full con-

fidence, *i.e.* $c_i = 1$ for all peaks $p_i \in \mathcal{S}$. Ideally, the estimated set $\hat{\mathcal{S}}$ should be of the same cardinality as $\mathcal{S}$ and the frequencies of each peak of $\hat{\mathcal{S}}$ should be close to the one of a peak in $\mathcal{S}$. By applying Algorithm 1, we iteratively seek for correspondences between peaks of the two sets by decreasing confidence order of estimated peaks until no peak remains in one of the two sets.

Algorithm 1 considered to seek for correspondence between peaks of the two sets $\mathcal{S}$ and $\hat{\mathcal{S}}$.

$c_I \leftarrow 0$
$\mathcal{S}_r \leftarrow \mathcal{S}$
$\hat{\mathcal{S}}_r \leftarrow \hat{\mathcal{S}}$
while $\mathcal{S}_r \neq \emptyset$ **and** $\hat{\mathcal{S}}_r \neq \emptyset$ **do**
  take $p_j \in \hat{\mathcal{S}}_r$ such that $c_j = \max_{p_k \in \hat{\mathcal{S}}_r} c_k$
  find $p_i \in \mathcal{S}_r$ such that $|f_i - f_j| = \min_{k \neq j} |f_k - f_j|$
  $c_I \leftarrow c_I + c(p_i, p_j)$
  $\mathcal{S}_r \leftarrow \mathcal{S}_r - \{p_i\}$
  $\hat{\mathcal{S}}_r \leftarrow \hat{\mathcal{S}}_r - \{p_j\}$
end while

Once the algorithm has been processed, the imprecision cost c_I is obtained. If the remaining set $\hat{\mathcal{S}}_r \neq \emptyset$, then some peaks have been over estimated. On contrary, if the remaining set $\mathcal{S}_r \neq \emptyset$, then some peaks have been under estimated. The over and under estimation costs are defined then by:

$$c_O = \sum_{j=1}^{\#\hat{\mathcal{S}}_r} c_j \quad \text{and} \quad c_U = \sum_{i=1}^{\#\mathcal{S}_r} c_i = \#\mathcal{S}_r \tag{13}$$

We propose that these three costs can be summed and normalized to obtain an overall cost that reflects both the issue of finding the correct number of peaks and of estimating their parameters precisely:

$$C = (c_O + c_U + c_I)/N \tag{14}$$

By using the cost function C, we consider that missing a sinusoid is equivalent to completely misestimating its frequency. In the same time, over estimating the number of sinusoids is a handicap only if the extractor had great confidence for the over detected peaks.

3.3. Test Cases

Now, the cost C defined in Equation 14 is used to assess the performance of a given extractor in a realistic framework. Two tests cases are considered to evaluate the precision and the resolution of the tested extractor, while the third one is intended for an overall evaluation by simulating a realistic case.

Precision. In this test, we consider only one sinusoid embedded in a Gaussian noise. Several runs with randomized frequencies and phases are operated and the mean cost versus the D-SNR is plotted such as in Figure 2. A perfect extractor will lead to performances equal to the CRB.

Resolution. To assess the resolution capabilities of a given extractor, we consider a test with two sinusoids with the same amplitude. Several runs with randomized phase and frequencies are operated and the mean cost versus the absolute difference between the frequencies of the two sinusoids will be plotted.

Overall Quality Assessment. In this test, we consider several sinusoids. Several runs with randomized frequencies and phases are operated and the mean cost versus the number of sinusoids will be plotted. For this test, the amplitudes may be set with the same value. This would be more correct, since it is much more difficult to detect a sinusoid of low amplitude than one with high amplitude and this is not reflected by the proposed cost C of Equation 14. However, tested estimators may take that into account to artificially achieve good evaluation. We then consider that the amplitude should also be randomized but within a reasonable range. For example, the threshold of hearing may be considered as a lower bound.

4. PARTIAL TRACKING

Trackers of partials are designed to identify continuities between peaks of (generally) successive frames to build a continuous representation of the evolutions of the sinusoidal parameters over time.

4.1. Existing Evaluation Methods

The first algorithms were proposed by McAulay and Quatieri in the speech processing field [12] and by Serra and Smith in the musical sound processing field [19]. They are now references and many implementations of these algorithms are available.

Several other methods have been proposed then [20, 21, 22, 23, 24, 25, 26] to achieve better identifications of the continuities between peaks either by extending these first algorithms or by considering a totally different framework.

Once proposed, most of the algorithms are validated by visual examination of the output of the trackers, *i.e.* the evolutions of the frequency or the amplitude of the partials are plotted over time. This may be used to show specific properties of a given algorithm but not to compare this algorithm to existing ones.

Alternatively, we have proposed and used a methodology of evaluation in [24, 25]. A single synthetic sinusoidal source x is corrupted by the addition of noise or another sinusoidal source y, leading to the signal s. The performance of the trusted tracker is evaluated by considering the ratio between the R-SNR (see Equation 7) versus the D-SNR (see Equation 11). The results for three trackers in two test cases are plotted in Figure 3. This methodology is useful to compare different trackers in a more systematic manner. However, since only SNRs are considered, it requires the implementation of the other stages of the chain, which could result in biased evaluations.

A first attempt to get rid of the influence of the other stages while assessing the performance of a tracker is proposed in [27]. By doing so, we have to compare the output of the tracker – set $\hat{\mathcal{L}}$ – to the original set $\mathcal{L}$.

In [26], Satar-Boroujeni *et al.* uses two factors defined as follows to compare the performance of their trackers with another one proposed in the literature:

$$R_d = n_d/n_e \quad \text{and} \quad R_f = n_f/n_e \tag{15}$$

where n_d is the number of detected tracks, n_f is the number of false tracks, and n_e is the number of expected tracks.

However, the direct comparison between the two sets of partials is a complicated task due to the high dimensionality of the elements of each set (a partial has a time location, and several parameters that evolve during this activity time slot), so that a comparison of the two sets by considering their cardinals appears as not sufficient. This issue will be addressed by the introduction of an algorithm to estimate how different two sets of partials are.

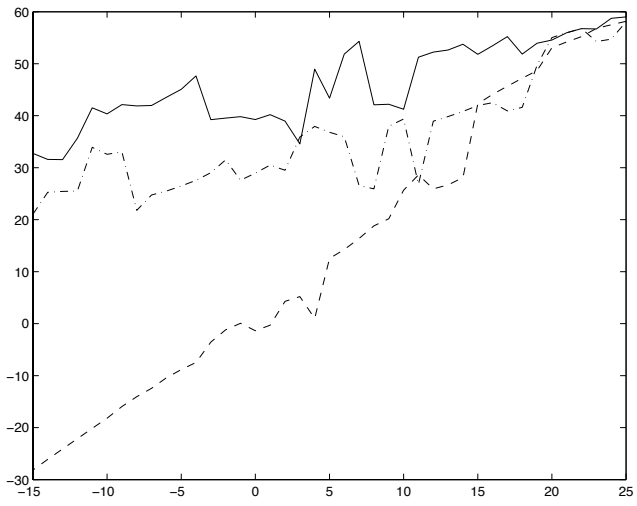


Figure 3: *Evaluation of the management of crossing partials for three trackers from [24]: the High-Frequency Content (HF) algorithm (solid line), the Linear-Prediction (LP) algorithm (dashed line), and the McAulay-Quatieri (MAQ) one (dash-dotted line). The R-SNR is plotted in function of the D-SNR.*

4.2. Evaluation Methodology

A set of partials $\mathcal{L}$ is considered as a reference. This set of partials $\mathcal{L}$ of cardinal $N = \#\mathcal{L}$ is converted into a set of peaks $\mathcal{S}$. After being degraded by several types of degradation studied in 4.3, this set is used as the input of the tested tracker to give another set of partials $\hat{\mathcal{L}}$ of cardinal $M = \#\hat{\mathcal{L}}$.

The issue is then to evaluate the quality of the tested tracker by comparing the two sets $\hat{\mathcal{L}}$ and $\mathcal{L}$.

4.2.1. Unitary Case

Let us first consider a simple case, where only one partial is in the original set and the estimated one: $N = M = 1$.

To evaluate how different two partials P_i and P_j are, we consider only the frequencies and amplitudes parameters. These parameters are supposed to be null if the partial is not active at a given frame:

$$A_i(k) = F_i(k) = 0 \text{ if } k < b_i \text{ or } k \geq b_i + l_i \tag{16}$$

(b_i being the birth index of partial i, and l_i its length).

Let us consider P_i as the reference partial. Considering that $b_i \leq b_j$, we define four frame indices as follows:

$$\begin{aligned} n_1 &= b_i \\ n_2 &= \min(b_j, b_i + l_i) \\ n_3 &= \min(\max(b_j, b_i + l_i), b_j + l_j) \\ n_4 &= \max(b_i + l_i, b_j + l_j) \end{aligned}$$

We assume that a missing part of the original partial has a cost equivalent to the cumulated amplitudes of the original partial. We also consider that artificially prolongating a partial in the resulting representation has a cost equivalent to the cumulative amplitude of the prolongated part.

The common part between the two partials (between frames n_2 and n_3) is then considered. We assume that the imprecision cost between a peak of the original partial p_i^k and a peak of the estimated partial $\hat{p}_j^k$ is the product of the differences between their frequencies and amplitudes.

The cost function between the two partials P_i and P_j is then:

$$\begin{aligned} c(P_i, P_j) &= \sum_{k=n_2}^{n_3} 2\,|F_i(k) - F_j(k)| \cdot |A_i(k) - A_j(k)| \\ &+ \sum_{k=n_1}^{n_2} \max(A_i(k), A_j(k)) + \sum_{k=n_3}^{n_4} \max(A_i(k), A_j(k)) \end{aligned} \tag{17}$$

Since we consider only real signals, the normalized frequencies are between 0 and 0.5. A normalization factor of 2 is then used to obtain a cost function between 0 and 1 (as in Equation 12).

4.2.2. General Case

Let us now consider the general case where N is set arbitrarily. Ideally, the estimated set $\hat{\mathcal{L}}$ should be of the same cardinality ($M = N$) and each partial of the estimated set should be close to one partial of $\mathcal{L}$. By applying Algorithm 2, we iteratively seek for correspondences between partials of the two sets until no partial remains in one of the two sets. This search is done by decreasing overall amplitude order of the estimated partials, the overall amplitude of a partial P_k being defined as:

$$A_k = \sum_{i=b_k}^{b_k+l_k-1} A_k(i) \tag{18}$$

Algorithm 2 considered to seek for correspondences between partials of two sets $\mathcal{L}$ and $\hat{\mathcal{L}}$.

$c_I \leftarrow 0$
$\mathcal{L}_r \leftarrow \mathcal{L}$
$\hat{\mathcal{L}}_r \leftarrow \hat{\mathcal{L}}$
while $\mathcal{L}_r \neq \emptyset$ **and** $\hat{\mathcal{L}}_r \neq \emptyset$ **do**
  take $P_j \in \hat{\mathcal{L}}_r$ such that $A_j = \max_{P_k \in \hat{\mathcal{L}}_r} A_k$
  find $P_i \in \mathcal{L}_r$ such that $c(P_i, P_j) = \min_{P_k \in \mathcal{L}_r} c(P_k, P_j)$
  $c_I \leftarrow c_I + c(P_i, P_j)$
  $\mathcal{L}_r \leftarrow \mathcal{L}_r - \{P_i\}$
  $\hat{\mathcal{L}}_r \leftarrow \hat{\mathcal{L}}_r - \{P_j\}$
end while

Once the algorithm has been processed, the imprecision cost c_I is obtained. If the remaining set $\hat{\mathcal{L}}_r \neq \emptyset$, then some partials have been over estimated. On the contrary, if the remaining set $\mathcal{L}_r \neq \emptyset$, then some partials have been under estimated. The over and under estimation costs are then defined by:

$$c_O = \sum_{j=1}^{\#\hat{\mathcal{L}}_r} A_j \quad \text{and} \quad c_U = \sum_{i=1}^{\#\mathcal{L}_r} A_i \tag{19}$$

We assume that these three costs can be summed and normalized by the number of partials of the original set $\mathcal{L}$ to obtain an overall cost that reflects the quality of the tracking:

$$C = (c_O + c_U + c_I)/N \tag{20}$$

4.3. Test Cases

The cost C defined in Equation 20 can be used to assess the performance of a given tracker. As described in the beginning of Section 4.2, a given set of partials $\mathcal{L}$ – considered as a reference – is converted back to a set of peaks $\mathcal{S}$. This set is possibly degraded to simulate degradations that trackers have to face when tracking partials within audio sounds.

4.3.1. Simulating Degradations

When considering a monophonic source embedded in noise or a polyphonic sound with many sources, the $\mathcal{S}$ set extracted by the first stage of the analysis / synthesis chain is usually degraded. In case of noise addition, the number of peaks is over estimated. In case of partials with close or crossing frequencies, some peaks are missing. In both cases, the parameters of the peaks are also not estimated precisely.

Therefore the three types of degradations proposed below – if combined – will produce a degradation close to the one obtained by the addition of a noise signal to the input signal $s(t)$ before processing a peak extractor.

We assume that these degradations may be considered independently in three different test cases.

Adding Peaks. In this test case, some peaks are added to $\mathcal{S}$ before the tracking. The frequencies and the phases of these peaks are randomly chosen. The amplitude of each peak is also chosen randomly, and the addition of peaks is stopped when the sum of their amplitudes has reached a given amount. The evaluation is done by considering the overall cost C versus the ratio between the summed amplitudes of the added peaks and the summed amplitudes of the peaks of $\mathcal{S}$.

Removing Peaks. In this test case, we randomly remove a given number of peaks from $\mathcal{S}$. The evaluation is done by considering the overall cost C versus the ratio between the number of removed peaks and the number of peaks in $\mathcal{S}$.

Degrading the Parameters of Partials. In this test case, the amplitude and frequency parameters of the peaks are modified by the addition of randomly chosen values. For the frequency, this value is chosen between $-\Delta_f$ and Δ_f. The evaluation is then done by considering the overall cost C versus the Δ_f threshold.

4.3.2. Polyphonic Case

In this last test case, we simulate a polyphonic context by adding several monophonic sound sources together within a unique set of partials before converting it to the peak representation $\mathcal{S}$.

However, this test is obviously over-simplified since the resulting representation is of far better quality than if the peak set was obtained by analyzing a polyphonic sound with a peak extractor.

5. CONCLUSION

In this paper, we have introduced two original evaluation methods that intend to ease the comparison of peak extractors and partial trackers in a realistic framework, and independently to any specific application. Clearly defined protocols are also proposed to evaluate most of the desired properties of peak extractors and partial trackers with several test cases that simulate typical problems while considering the sinusoidal model in audio processing applications. Each protocol allows to evaluate individually each stage of the sinusoidal analysis / synthesis chain.

Although the capacities of extractors to detect the sinusoids and to estimate the parameters of these sinusoids are generally evaluated independently, they are in practice inextricably linked. The proposed method for peak extractors uses a unique cost function that globally reflects the performance of the tested method.

The protocol for partial trackers considers a cost function that evaluates how different the extracted set of partials is from the original one. The definition of this cost function is a difficult problem, since the comparison of two sets of partials is not clearly defined yet. The empirical cost function proposed in this paper constitutes a first attempt to address this issue. However, a more theoretically motivated comparison method should be considered as a future research subject.

This paper is a first proposal to be used as a starting point in a sinusoidal analysis / synthesis contest to be held at DAFx'07. Participants will have the opportunity to submit extractors, trackers, as well as evaluation methodologies and protocols. The resulting survey will be automatically generated and published. This survey should be of great interest for people involved in sinusoidal modeling.

6. REFERENCES

[1] Ircam, *AudioSculpt User's Manual*, 2nd ed., Ircam, Paris, 1996.

[2] K. Fitz and L. Haken, "Sinusoidal modeling and manipulation using Lemur," *Computer Music J.*, vol. 20, no. 4, pp. 44–59, 1996.

[3] X. Serra, *Musical Signal Processing*, ser. Studies on New Music Research. C. Roads, S. T. Pope, A. Piccialli, G. De Poli (eds.), Swets & Zeitlinger, Lisse, the Netherlands, 1997, ch. Musical sound modeling with sinusoids plus noise, pp. 91–122.

[4] S. Marchand and R. Strandh, "InSpect and ReSpect: spectral modeling, analysis and real-time synthesis software tools for researchers and composers," in *Proc. Int. Comp. Music Conf. (ICMC'99),* Beijing, China, 1999, pp. 341–344.

[5] R. J. McAulay and T. F. Quatieri, "Shape invariant time-scale and pitch modification of speech," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 40, no. 3, pp. 497–510, 1992.

[6] S. N. Levine, T. S. Verma, and J. O. Smith, "Multiresolution sinusoidal modeling for wideband audio with modifications," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'98),* Seattle, USA, vol. 6, 1998, pp. 3585–3588.

[7] H. Purnhagen, N. Meine, and B. Edler, "Sinusoidal coding using loudness-based component selection," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'02),* Orlando, USA, vol. 2, 2002, pp. 1817–1820.

[8] G. Garcia and J. Pampin, "Data compression of sinusoidal modeling parameters based on psychoacoustic masking," in *Proc. Int. Comp. Music Conf. (ICMC'99),* Beijing, China, 1999, pp. 40–43.

[9] F. Keiler and S. Marchand, "Survey on extraction of sinusoids in stationary sounds," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-02),* Hamburg, Germany, 2002, pp. 51–58.

[10] S. W. Hainsworth and M. D. Macleod, "On sinusoidal parameter estimation," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-03),* London, UK, 2003, pp. 151–156.

[11] R. Badeau, B. David, and G. Richard, "High resolution spectral analysis of mixtures of complex exponentials modulated by polynomials," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 54, no. 4, pp. 1341–1350, 2006.

[12] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 34, no. 4, pp. 744–754, 1986.

[13] L. Girin, S. Marchand, J. di Martino, A. Röbel, and G. Peeters, "Comparing the order of a polynomial phase model for the synthesis of quasi-harmonic audio signals," in *Proc. IEEE Workshop Appl. of Dig. Sig. Proc. to Audio and Acoust.,* New Palz, NY, 2003, pp. 193–196.

[14] A. Freed, X. Rodet, and P. Depalle, "Performance, synthesis and control of additive synthesis on a desktop computer using FFT^{-1}," in *Proc. Int. Comp. Music Conf. (ICMC'93),* Tokyo, Japan, 1993, pp. 98–101.

[15] N. Meine and H. Purnhagen, "Fast sinusoid synthesis for MPEG-4 HILN parametric audio decoding," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-02),* Hamburg, Germany, 2002, pp. 105–110.

[16] CRC, "CRC-Seaq," [Online] http://www.crc.ca/fr/html/aas/home/products/products, 2000.

[17] D. C. Rife and R. R. Boorstyn, "Single-tone parameter estimation from discrete-time observations," *IEEE Trans. Information Theory*, vol. IT-20, pp. 591–598, 1974.

[18] M. Lagrange and S. Marchand, "Improving sinusoidal frequency estimation using a trigonometrical approach," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-05),* Madrid, Spain, 2005, pp. 59–64.

[19] J. O. Smith and X. Serra, "PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *Proc. Int. Comp. Music Conf. (ICMC'87),* Champaign-Urbana, USA, 1987, pp. 290–297.

[20] P. Depalle, G. Garcia, and X. Rodet, "Tracking of partials for additive sound synthesis using Hidden Markov Models," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'93),* Minneapolis, USA, 1993, pp. 225–228.

[21] A. Sterian and G. H. Wakefield, "A model-based approach to partial tracking for musical transcription," in *Proc. SPIE Int. Symp. Optical Science, Eng., and Instrumentation, San Diego, California*, San Diego, 1998.

[22] H. Purnhagen, "Parameter estimation and tracking for time-varying sinusoids," in *Proc. 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA)*, Leuven, Belgium, 2002.

[23] M. Lagrange, S. Marchand, and J.-B. Rault, "Partial tracking based on future trajectories exploration," in *116th Conv. Audio Eng. Soc.,* Berlin, Germany, 2004, preprint 6046 (10 pages).

[24] ——, "Using linear prediction to enhance the tracking of partials," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'04),* Montreal, Canada, vol. 4, 2004, pp. iv/241 – iv/244.

[25] ——, "Tracking partials for the sinusoidal modeling of polyphonic sounds," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'05),* Philadelphia, USA, vol. 3, 2005, pp. iii/229 – iii/232.

[26] H. Satar-Boroujeni and B. Shafai, "A robust algorithm for partial tracking of music signals," in *Proc. Int. Conf. on Digital Audio Effects (DAFx-05),* Madrid, Spain, 2005, pp. 202–207.

[27] M. Lagrange, "Sinusoidal modeling of polyphonic sounds," Ph.D. dissertation, University of Bordeaux 1, France, 2004, in French.