ENHANCED PARTIAL TRACKING USING LINEAR PREDICTION

Mathieu Lagrange†, Sylvain Marchand‡, Martin Raspaud‡, and Jean-Bernard Rault†

†France Telecom R&D
4, rue du Clos Courtel, BP 59
F-35512 Cesson Sevigné cedex, France
firstname.name@rd.francetelecom.com

‡SCRIME – LaBRI, Université Bordeaux 1
351, cours de la Libération,
F-33405 Talence cedex, France
[sm|mraspaud]@labri.fr

ABSTRACT

In this paper, we introduce a new partial tracking method suitable for the sinusoidal modeling of mixtures of instrumental sounds with pseudo-stationary frequencies. This method, based on the linear prediction of the frequency evolutions of the partials, enables us to track these partials more accurately at the analysis stage, even in complex sound mixtures. This allows our spectral model to better handle polyphonic sounds.

1. INTRODUCTION

Spectral sound models provide general representations for many applications such as compression, content extraction and transformation. Most of these models, such as additive synthesis, are based on the Fourier analysis which has proven to be accurate under the condition of local stationarity.

Since the Fourier analysis delivers a short-time spectral representation of the analyzed sound, we consider maxima in the magnitude spectrum (so-called peaks) to be the instantaneous representation of partials. We have then to link peaks of successive frames to recover the continuous evolution of the partials.

After a brief introduction in Section 2 to the model used in this paper, we present the partial tracking algorithm introduced by McAulay and Quatieri in [1] and few extensions proposed to improve the synthesis quality in Section 3. Section 4 is dedicated to the presentation of the linear prediction (LP) model, where three methods for the estimation of the LP coefficients are compared. Given this model, we propose a new tracking method in Section 5 which takes advantage of the past evolutions of the partials to predict their optimal future evolution. We discuss the choice of the method used to estimate the LP coefficients (using both synthetic and natural sounds), then we give an overview of the algorithm strategy in Section 6.

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*. Moreover, we assume that these evolutions are predictable since sudden changes in the evolutions of the partials will generate noisy "clicks". The audio signal s can be calculated from the additive parameters using Equations 1 and 2, where P is the number of partials and the functions f_p, a_p, and ϕ_p are the instantaneous frequency, amplitude, and phase of the p-th partial, respectively. The P pairs (f_p, a_p) 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 Lemur [2], SMS [3], or InSpect [4].

$$s(t) = \sum_{p=1}^{P} a_p(t)\, \cos(\phi_p(t)) \quad (1)$$

$$\phi_p(t) = \phi_p(0) + 2\pi \int_0^t f_p(u)\, du \quad (2)$$

3. MC AULAY AND QUATIERI ALGORITHM

The first partial tracking algorithm was introduced in [1] by McAulay and Quatieri in the field of the sinusoidal modeling of the voice. In this section we present the basic algorithm, proposed extensions, and its limitations.

3.1. Basic Algorithm

This algorithm follows a short-term analysis, where short-time spectra are extracted from the sound at different time frames. The algorithm is based on the assumption that partials composing a voiced signal have stationary frequency evolutions. It is then proposed to consider frequency differences between spectral peaks of immediately successive frames to form partials. A maximal frequency difference threshold Δ_f between successive peaks of a partial is set:

$$|f_p(k+1) - f_p(k)| < \Delta_f \quad (3)$$

where $f_p(k)$ is the frequency of the p-th partial at frame k.

The algorithm is processed iteratively frame by frame and by increasing frequency (partials having peaks with low frequencies are linked first). Suppose that all peaks of frames having indices below k are processed. For a peak ρ_i^k of index i within frame k, we look for an unlinked peak ρ_j^{k+1} so that the frequency difference between those peaks is minimal. If the frequency difference is greater than Δ_f, the current partial is labeled "dead". If not, ρ_j^{k+1} is reserved.

If this peak cannot be better linked with the next peak ρ_{i+1}^k, the partial having peak ρ_i^k definitively links peak ρ_j^{k+1}. If not, the current partial looks for another candidate in the next frame. If no alternative can be found, the partial is also labeled "dead". After all reachable peaks of frame $k+1$ are linked, unlinked peaks of frame $k+1$ give rise to new partials.

3.2. Extensions

It was proposed to add to each partial at the beginning and at the end of the track a zero-amplitude peak (frequency and phase being extrapolated) to improve re-synthesis quality. It is particularly suitable for the modeling of voice signals but can lead to "pre-echo" artifacts in the case of non stationary signal in the modeling of musical signals.

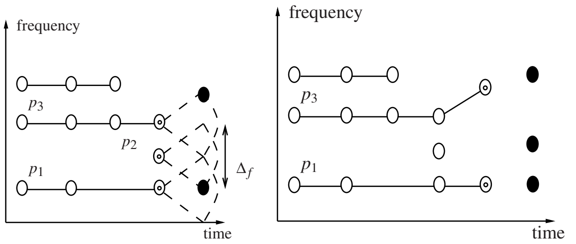

Figure 1: *One step of the McAulay-Quatieri algorithm. White peaks cannot be candidate, whereas black peaks can. Peaks with circles are "heads" of partials.*

In the original algorithm, partials are processed in increasing frequency. Since partials of great amplitude are more perceptively important, it was proposed to process them first to avoid discontinuities leading to noisy "clicks" that can be heard at the synthesis stage.

For various reasons such as decreasing amplitude, strong modulations or DFT bin corruption, a peak selection process can remove erroneous peaks [5], but is not able to recover the underlying spectral information. It leads to a lack of peaks. To overcome this analysis drawback, it is proposed in [3] to add a "zombie" state to the partials, so that if a partial cannot link to any peak in a frame, it can still look for its next peak in a limited number of frames. If a peak can be found, the missing parameters of "zombie" peaks are then interpolated.

3.3. Limitations

These extensions (allowing better re-synthesis quality for given analyzed signal types) are mainly algorithmic improvements. Indeed, the basis of the algorithm (the fact that the frequency and amplitude trajectories are considered as constant) is left untouched. However, for the frequency, this assertion is rarely verified since for many musical instruments including singing voice, vibrato or portamento are commonplace; and for the amplitude, the variations are so strong that it is nearly never verified.

Furthermore, in the "zombie" extension, since the partial frequency is considered as constant – stationary – from one frame to another, the repeated use of the "zombie" state often makes the trajectory of the partial to diverge from the real – non stationary – one.

We aim at taking into account this non-stationary evolution by considering that the parameters of the partials are no more close to the past value but close to a linear combination of past values.

4. LINEAR PREDICTION

In the linear prediction (LP) model, also known as the autoregressive (AR) model, the current sample $x(n)$ is approximated by a linear combination of past samples of the input signal. We are then looking for a vector a of d coefficients, d being the order of the LP model. Provided that the a vector is estimated, the predicted value $\hat{x}$ is computed simply by FIR filtering of the p past samples with the coefficients using Equation 4:

$$\hat{x}(n) = \sum_{i=1}^{p} a_i x(n-i) \quad (4)$$

The main challenge in linear prediction modeling is to use a method for the estimation of the coefficients that suits specific needs. Three methods used to estimate the a vector and their specific requirements and capabilities are presented. The autocorrelation and covariance methods are detailed in [6]. The autocorrelation and Burg methods are presented on both theoretical and computational points of view in [7].

4.1. Autocorrelation Method

The autocorrelation method minimizes the forward prediction error power on an infinite support, that is:

$$\varepsilon_{\infty}^{f} = \frac{1}{N} \sum_{\infty}^{-\infty} |x(n) - \hat{x}(n)|^2 \quad (5)$$

where $\hat{x}(n)$ is the estimate computed with Equation 4. Since the signal is finite, samples of the $x(n)$ process which are not observed are then set to zero and observed samples are windowed in order to minimize the discontinuity at the boundaries. Finding the coefficients by minimizing ε_{∞}^{f} leads us to solve a very regular system of normal equations – a Toeplitz matrix – efficiently solved by the Levinson-Durbin algorithm.

4.2. Covariance Method

On contrary, the covariance method assumes finite support for the minimization of the forward prediction error power:

$$\varepsilon^{f} = \frac{1}{(N-p)} \sum_{N-1}^{p} |x(n) - \hat{x}(n)|^2 \quad (6)$$

where $\hat{x}(n)$ is the estimate computed with Equation 4. Since no zeroing of the data is necessary, this method is a good candidate for coefficients estimation of process having few observed samples. Unfortunately the method can lead to filters that are not minimal phase (the estimated poles are not guaranteed to lie within the unit circle, i.e not all $a_i < 1$).

4.3. Burg Method

Let $e_k^f(n)$ and $e_k^b(n)$ respectively denote the forward and backward prediction errors for a given order k:

$$e_k^f(n) = x(n) + \sum_{i=1}^{k} a(i) x(n-i) \quad (7)$$

$$e_k^b(n) = x(n-k) + \sum_{i=1}^{k} a(i) x(n+k+i) \quad (8)$$

The Burg method minimizes the average of the forward and backward error power on a finite support in a recursive manner. That is, to obtain $a(k)$ we minimize:

$$\varepsilon_k = \frac{1}{2}(\varepsilon_k^f + \varepsilon_k^p) \quad (9)$$

where

$$\varepsilon_k^f = \frac{1}{(N-k)} \sum_{n=k}^{N-1} |e_k^f(n)|^2 \quad (10)$$

$$\varepsilon_k^b = \frac{1}{(N-k)} \sum_{n=0}^{N-1-k} |e_k^b(n)|^2 \quad (11)$$

and

$$a_k(i) = \begin{Bmatrix} a_{k-1}(i) + r_k a_{k-1}(k-i) \text{ for } i = 1, 2, ..., k-1 \\ r_k \text{ for } i = k \end{Bmatrix} \quad (12)$$

where r_k is called the reflection coefficient. By substituting Equation 12 in Equations 10 and 11, we find a recursive expression for the forward and backward errors:

$$e_k^f(n) = e_{k-1}^f(n) + r_k e_{k-1}^b(n-1) \tag{13}$$

$$e_k^b(n) = e_{k-1}^b(n-1) + r_k e_{k-1}^f(n) \tag{14}$$

where

$$e_0^f(n) = e_0^b(n) = x(n) \tag{15}$$

To find r_k, we differentiate the k^{th} prediction error power with respect to r_k and by setting the derivative to zero, we obtain:

$$r_k = \frac{-2\sum_{n=k}^{N-1} e_{k-1}^f(n) e_{k-1}^b(n-1)}{\sum_{n=k}^{N-1} |e_{k-1}^f(n)|^2 + |e_{k-1}^b(n-1)|^2} \tag{16}$$

The Burg method combines advantages of both previous methods. As the autocorrelation method, the Burg method is minimal phase ($\forall i, a_i < 1$). And as the covariance method, the Burg method estimates the a_i on a finite support.

The following algorithm computes the vector a of linear prediction coefficients using the Burg method, at the order d:

ef $\leftarrow x$
eb $\leftarrow x$
$a \leftarrow 1$
for m from 0 to $d-1$ do
  efp $\leftarrow$ ef without its first element
  ebp $\leftarrow$ eb without its last element
  $k \leftarrow -2\text{ebp}\cdot\text{efp}/(\text{ebp}\cdot\text{ebp}+\text{efp}\cdot\text{efp})$
  ef $\leftarrow$ efp $+ k$ ebp
  eb $\leftarrow$ ebp $+ k$ efp
  $a \leftarrow (a[0], a[1], \cdots, a[m], 0) + k(0, a[m], a[m-1], \cdots, a[0])$
end for

5. PREDICTION OF THE EVOLUTION OF PARTIALS

As shown in [8, 9] by Kauppinen *et al*, linear prediction can be successfully used to extrapolate audio signals. We show here that it can also be used for predicting the evolutions of the partials in time, which are time signals too – with a much lower sampling frequency though.

The methodology used in order to estimate the next frequency value leads to an enhanced tracking algorithm.

5.1. Prediction Algorithm

To obtain an estimation of the future evolution of a partial, we use samples of its past evolution. The number of samples considered is in the $[1, N_s]$ range, where N_s is the maximal number of samples to be considered. When the number of samples is too small with respect to the model order – at the beginning of the partial – we use the following reflection algorithm.

Considering a vector x of size N, the left reflected values are: $x(-i) = 2x(0) - x(i)$ and symmetrically the right reflected values are: $x(N+i) = 2x(N) - x(i)$. This method conserves both zero and first order continuities.

An estimation method is then applied on those samples to find out LP coefficients. These coefficients are then used with Equation 4 to obtain the predicted frequency of the next peak.

5.2. LP Coefficients Estimation Method

In Section 4, we introduced three methods to estimate the LP coefficients. To choose the method that best suits our needs, we compare the three methods on their ability to predict synthetic signal close to natural signals.

Before computing the LP coefficients using the autocorrelation method, the signal was zero-centered by subtracting its mean. We can see in Figure 2 that the autocorrelation method performs badly on short data tracks. Since the parameters of the partials are sampled at a very low frequency, it seems unacceptable to use this method. The covariance method performs better but can be unstable in the presence of bursting noise (see Figure 2 at frame 32). This instability can lead to erroneous links in the presence of noise. The Burg method seems to be a good compromise in terms of reactivity and resistance to noise.

5.3. LP Parameters

The model order and the number of samples used to estimate the LP coefficients are of great importance for the quality of the prediction. We choose parameter range values by both theoretical and experimental considerations.

Our experimental tests were processed on the known evolutions of frequencies of already-tracked partials of different kinds, of pseudo-stationary monophonic signals such as a saxophone, a guitar, and different singing voices. We considered the mean prediction error and the maximal error.

For frequency evolutions, since we want to model exponentialy increasing or decreasing evolutions (portamento) and sinusoidal evolutions (vibrato), the order of the LP model should not be below 2. Experimental testing showed that a model order in the $[2, 8]$ range is convenient.

The number of samples used has to be large enough to be able to extract the signal periodicity, and short enough not to be too constrained by the past evolution. The short-term analysis module uses a sliding time / frequency transform with a hop size of 512 samples on sound signals sampled at CD quality (44.1 kHz). This means that the frequency and amplitude trajectories are sampled at ≈ 86 Hz. Since we want to handle natural vibrato with a frequency about 4 Hz, we need at least 20 samples to get the period of the vibrato. Experimental testing showed that for most cases a number of samples in the $[4, 32]$ range is convenient.

5.4. Prediction Results

In Tables 1 and 2, we present experimental test results of the LP predictor with several orders and numbers of samples on already-tracked partials of natural sounds: a saxophone and a singing voice (one harmonic of each sound is plotted on Figure 3 and 4). Additionally, we compare it with several simple predictors, the constant predictor, used in the extended McAulay-Quatieri algorithm, the linear predictor and the reflection one, respectively defined as:

- *Constant predictor*: $\hat{x}(n+k) = x(n)$,
- *Linear predictor*: $\hat{x}(n+k) = x(n) + (x(n) - x(n-1)) \cdot k$,
- *Reflection predictor*: $\hat{x}(n+k) = 2x(n) - x(n-k)$

where k is the distance in frames between the last observation and the predicted value. The mean and the maximum (in parentheses on Tables 1 and 2) of the prediction error are considered. Prediction errors are computed for several simple predictors presented above (left part) and the LP predictor (right part) for different values of k. Concerning the part dedicated to the LP predictor, the model order grows from left to right, and for each values of k the number of samples considered is $[4, 8, 16, 32]$.

The LP model is efficient for constant, sinusoidal and exponentially increasing or decreasing processes. Concerning constant evolutions (frequency evolution of partials of a piano tone), the constant predictor is sufficient. Concerning sinusoidal evolutions – vibrato in the time/frequency plane – (see Figure 1), providing that the number of samples is sufficient to get the vibrato period, the LP predictor shows great results both in mean and maximal errors especially when k is increasing. For exponentially increasing

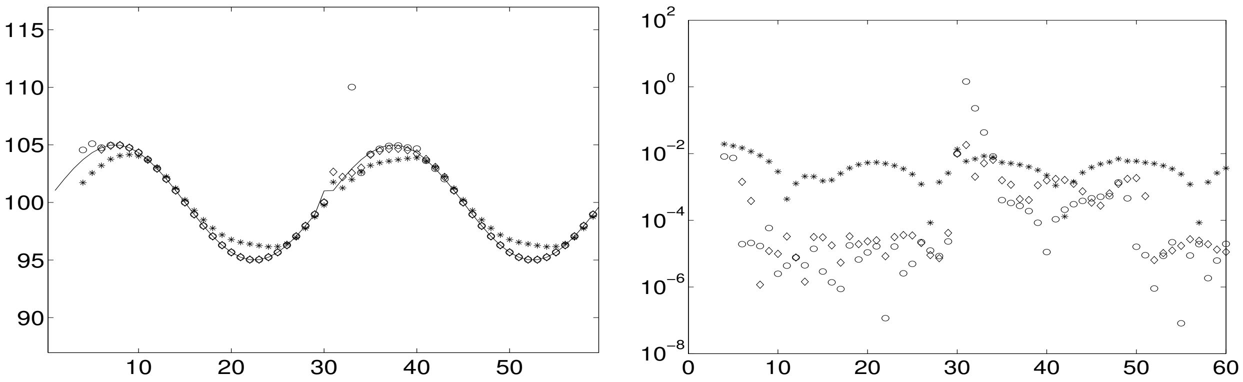

Figure 2: *On the left, the extrapolation at different times of one value in the future with a maximum of 12 previous samples and a 4-order predictor using three linear prediction methods: correlation (∗), covariance (○), and Burg (◇). The original (solid line) is a periodic signal with one sample – in the middle – displaced by a multiplication by a factor 1.01 (for clarity sake, some diverging covariance samples are not plotted). On the right, the evolutions of the associated prediction errors are displayed.*

or decreasing evolutions – portamento in the time/frequency plane – (see Figure 2), the mean error of the LP predictor is good comparing to simple predictors, but the maximum error is comparable, mainly due to the unpredictability of a non-stationary transition (see Figure 4 at frame 85).

6. ENHANCED TRACKING ALGORITHM

The algorithm has a general structure that can be divided in two steps: scheduling and reservation. The scheduling process consists in sorting the partials in decreasing overall amplitude. The overall amplitude of a partial is defined as the sum of the amplitudes of all the peaks included in the partial. Given a linear prediction modeling of the past evolution of the partial in the time / frequency plane, we are able to predict the next frequency value that leads to the smoothest frequency evolution for each partial. The reservation step consists in choosing the peak in the next frame that has the frequency nearest to the predicted one, and reserving it if the absolute difference between the frequency of the selected peak and the predicted frequency is below a given threshold. This rejection threshold is similar as Δ_f (see Equation 3) used in the McAulay-Quatieri algorithm, but it can be set to a much smaller value. This allows us to better avoid the selection of erroneous peak candidates, thus increasing the robustness of the partial tracking algorithm with polyphonic sounds.

For each partials, those two steps are processed until all reachable peaks of next frame are linked. Unlinked peaks of frame $k+1$ then give rise to new partials, and for partials in the "zombie" state we add an extrapolated "zombie" peak with the amplitude and frequency values predicted from the past evolutions. The "zombie" state can be used only for a limited number of consecutive frames (set to 4 in our algorithm). Beyond this limit, the partial is labeled "dead".

7. CONCLUSION AND FUTURE WORK

In this paper, we have considered the non-stationarity of the parameters of a sinusoidal model and we have taken advantage of linear prediction in order to forecast the evolutions of the partials in both frequency and amplitude, thus leading to an enhanced partial tracking algorithm. These considerations greatly improve the accuracy of spectral modeling.

Of course, the results presented here are still preliminary. Although we have only presented the results for the evolutions in frequency, the same method works also very well for the evolutions in amplitude. A comparison with other partial tracking algorithms such as [10] for example has to be done in the near future.

8. REFERENCES

[1] Robert J. McAulay and Thomas F. Quatieri, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," *IEEE ICASSP*, vol. 34, no. 4, pp. 744–754, 1986.

[2] Kelly Fitz and Lippold Haken, "Sinusoidal Modeling and Manipulation Using Lemur," *Computer Music Journal*, vol. 20, no. 4, pp. 44–59, Winter 1996.

[3] Xavier Serra, *Musical Signal Processing*, chapter Musical Sound Modeling with Sinusoids plus Noise, pp. 91–122, Studies on New Music Research. Swets & Zeitlinger, Lisse, the Netherlands, 1997.

[4] Sylvain Marchand and Robert Strandh, "InSpect and ReSpect: Spectral Modeling, Analysis and Real-Time Synthesis Software Tools for Researchers and Composers," in *Proc. ICMC*, Beijing, China, October 1999, ICMA, pp. 341–344.

[5] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault, "Sinusoidal Parameter Extraction and Component Selection in a Non Stationary Model," in *Proc. DAFx*. University of the Federal Armed Forces, Hamburg, September 2002, pp. 59–64.

[6] John Makhoul, "Linear Prediction : A Tutorial Review," *Proceedings of the IEEE*, vol. 63, no. 4, November 1992.

[7] Florian Keiler, Daniel Arfib, and Udo Zölzer, "Efficient Linear Prediction for Digital Audio Effects," in *Proc. DAFx*. Università degli Studi di Verona and COST, December 2000.

[8] Ismo Kauppinen, Jyrki Kauppinen, and Pekka Saarinen, "A Method for Long Extrapolation of Audio Signals," *JAES*, vol. 49, no. 12, pp. 1167–1180, December 2001.

[9] Ismo Kauppinen and Kari Roth, "Audio Signal Extrapolation – Theory and Applications," in *Proc. DAFx*. University of the Federal Armed Forces, Hamburg, September 2002, pp. 105–110.

[10] Philippe Depalle, Guillermo Garcia, and Xavier Rodet, "Analysis of Sound for Additive Synthesis: Tracking of Partials Using Hidden Markov Models," in *Proc. ICMC*, San Francisco, 1993, ICMA.

k:	Constant	Linear	Reflection	order : 2	4	6	8
	0.35 (0.9)	0.16 (0.8)	0.16 (0.8)	0.20 (0.8)	- (-)	- (-)	- (-)
1	- (-)	- (-)	- (-)	0.17 (0.6)	0.17 (0.6)	0.18 (0.7)	- (-)
	- (-)	- (-)	- (-)	0.16 (0.6)	0.14 (0.6)	0.14 (0.6)	0.14 (0.6)
	- (-)	- (-)	- (-)	0.16 (0.7)	0.13 (0.6)	0.13 (0.7)	0.12 (0.6)
	0.69 (1.8)	0.42 (1.4)	0.51 (1.6)	0.48 (1.4)	- (-)	- (-)	- (-)
2	- (-)	- (-)	- (-)	0.43 (1.3)	0.42 (1.3)	0.45 (1.6)	- (-)
	- (-)	- (-)	- (-)	0.41 (1.3)	0.35 (1.3)	0.34 (1.4)	0.34 (1.3)
	- (-)	- (-)	- (-)	0.41 (1.3)	0.32 (1.2)	0.29 (1.1)	0.28 (1.1)
	1.01 (2.5)	0.76 (2.4)	1.00 (3.1)	0.85 (2.3)	- (-)	- (-)	- (-)
3	- (-)	- (-)	- (-)	0.75 (2.3)	0.77 (2.3)	0.80 (2.7)	- (-)
	- (-)	- (-)	- (-)	0.72 (2.2)	0.62 (2.0)	0.57 (2.2)	0.57 (2.1)
	- (-)	- (-)	- (-)	0.71 (2.1)	0.52 (1.7)	0.46 (1.7)	0.43 (1.4)
	1.31 (3.1)	1.18 (3.7)	1.63 (4.6)	1.29 (3.5)	- (-)	- (-)	- (-)
4	- (-)	- (-)	- (-)	1.13 (3.5)	1.18 (3.6)	1.20 (4.1)	- (-)
	- (-)	- (-)	- (-)	1.09 (3.5)	0.92 (3.3)	0.83 (3.1)	0.81 (3.0)
	- (-)	- (-)	- (-)	1.06 (3.3)	0.77 (2.5)	0.65 (2.3)	0.58 (1.9)

Table 1: *The mean and (maximal) prediction errors for different predictors on the frequency evolution of partials of* **Saxophone Vibrato**. *Prediction errors are computed for several simple predictors (left part) and the LP predictor (right part) for different values of k (the distance in frame indices between the last observation and the predicted value). Concerning the part dedicated to the LP predictor, the model order grows from left to right, and for each values of k the number of samples considered is in* $[4, 8, 16, 32]$. *The prediction errors of the LP predictor are lower than those of the best simple predictor. The improvement is getting more and more significant when k is increasing.*

k:	Constant	Linear	Reflection	order : 2	4	6	8
	2.69 (26.3)	1.16 (10.6)	1.16 (10.6)	1.39 (10.1)	- (-)	- (-)	- (-)
1	- (-)	- (-)	- (-)	1.22 (10.0)	0.97 (12.2)	1.09 (11.5)	- (-)
	- (-)	- (-)	- (-)	1.19 (9.8)	0.86 (12.4)	0.89 (11.4)	0.92 (11.3)
	- (-)	- (-)	- (-)	1.17 (9.6)	0.83 (12.4)	0.84 (11.1)	0.86 (11.2)
	5.32 (44.9)	3.19 (30.6)	3.99 (36.8)	3.66 (29.8)	- (-)	- (-)	- (-)
2	- (-)	- (-)	- (-)	3.24 (28.7)	2.67 (35.1)	2.98 (33.1)	- (-)
	- (-)	- (-)	- (-)	3.17 (27.7)	2.35 (35.2)	2.35 (33.3)	2.43 (33.9)
	- (-)	- (-)	- (-)	3.12 (26.7)	2.24 (34.4)	2.24 (32.5)	2.26 (33.0)
	7.81 (61.2)	6.01 (56.8)	8.06 (56.8)	6.61 (55.2)	- (-)	- (-)	- (-)
3	- (-)	- (-)	- (-)	5.88 (53.0)	4.96 (62.2)	5.39 (61.3)	- (-)
	- (-)	- (-)	- (-)	5.76 (51.0)	4.33 (61.8)	4.21 (60.8)	4.32 (61.6)
	- (-)	- (-)	- (-)	5.68 (49.1)	4.10 (59.2)	4.00 (57.9)	3.99 (58.9)
	10.17 (74.9)	9.45 (84.5)	12.76 (72.7)	10.02 (81.6)	- (-)	- (-)	- (-)
4	- (-)	- (-)	- (-)	8.91 (78.0)	7.60 (88.6)	8.18 (86.1)	- (-)
	- (-)	- (-)	- (-)	8.77 (74.9)	6.64 (86.8)	6.40 (85.1)	6.51 (85.5)
	- (-)	- (-)	- (-)	8.67 (71.9)	6.29 (81.5)	6.07 (79.6)	6.00 (80.3)

Table 2: *The mean and (maximal) prediction errors for different predictors on the frequency evolution of partials of* **Singing Voice**. *Prediction errors are computed for several simple predictors (left part) and the LP predictor (right part) for different values of k (the distance in frame indices between the last observation and the predicted value). Concerning the part dedicated to the LP predictor, the model order grows from left to right, and for each values of k the number of samples considered is in* $[4, 8, 16, 32]$. *The mean prediction error of the LP predictor is lower than those of the best simple predictor, and the improvement is getting more and more significant when k is increasing. The maximum error remains comparable, mainly due to the unpredictability of a non-stationary transition (see Figure 4 at frame 85).*

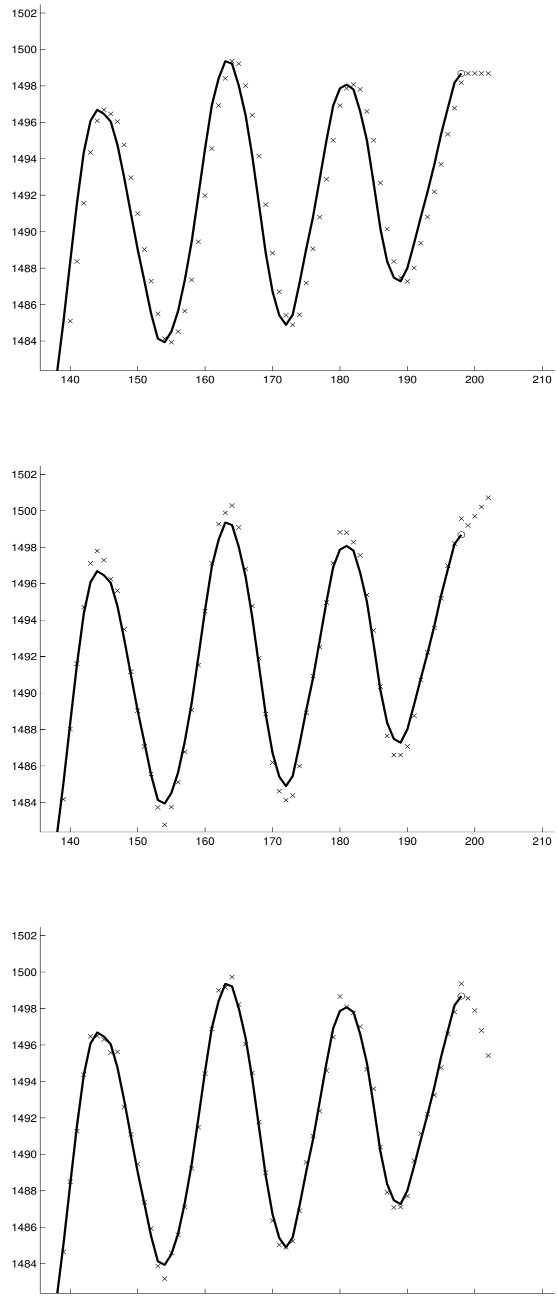

Figure 3: *Evolutions of different predictors: constant (top), linear (middle), LP (bottom) for* **Saxophone Vibrato***. The order of the LP predictor is 6 and the number of past samples used to compute the coefficients is 20. The frequency evolutions of the partials are drawn with lines and the end of the partial is represented by a circle. The predicted frequencies, plotted with crosses, are computed using the last observed values. To show the ability of the predictor to extrapolate at a longer term (in case of missing peaks), the predicted values are plotted even after the death of the partial.*

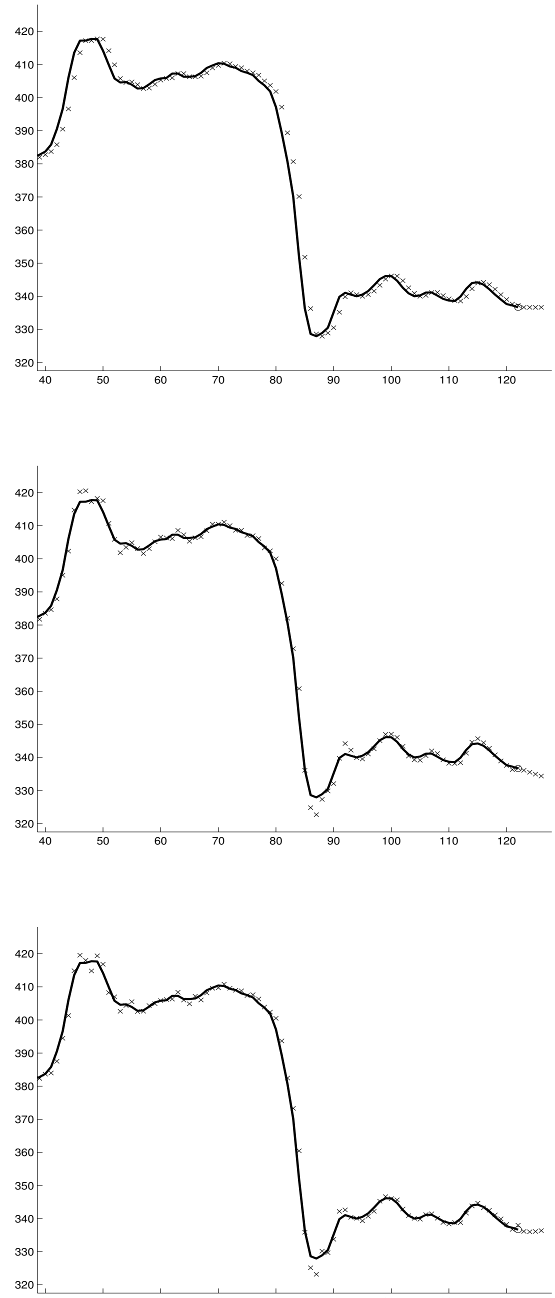

Figure 4: *Evolutions in time of different predictors: constant (top), linear (middle), LP (bottom) for* **Singing Voice***. The order of the LP predictor is 6 and the number of past samples used to compute the coefficients is 20.*