# PARTIAL TRACKING IN TWO STEPS

*Ádám Siska*

sales@sadam.hu

## ABSTRACT

This paper proposes a new algorithm for the creation of spectral envelopes in the context of additive analysis of recorded sounds. The key idea behind this method is the recognition, that the behaviour of a partial is different on the microscopic and the macroscopic level. Thus, our suggested process divides the envelope matching procedure into a micro-level and a macro-level analysis. Our current results are promising, however, further investigation is needed to understand every detail of this method.

## 1. INTRODUCTION

Johannes Kretz's KLANGPILOT 3 environment [2] is a real-time Computer Aided Composition tool that addresses the issue of including spectral qualities (amongst them, synthesis parameters) in a composition – particularly, in its score. This software has an extensive synthesis module allowing an arbitrary combination of additive, subtractive and FOF (fonction d'onde formantique) synthesis methods.

As the synthesis possibilities of the system started to grow, efficient ways had to be developed to ease the composers' task of virtual instrument creation. It was a plausible idea to build an additive analysis module which would create additive synthesis-based instruments that the users could modify according to their own purposes. During the implementation of this module, we realized that every partial tracking principle accessible for us was based on the same method of taking an initial number of tonal peaks and trying to extend them as long as possible [1, 3, 5, 8, 10]. Our own experience is, on the contrary, that partials won't share the same aspects on the macroscopic scale of musically perceivable durations than on the microscopic level of a few milliseconds. This led us to the development of a new method for partial tracking, which we present in this article.

The main difference between these aspects lies in predictability. On very short-terms it is not hard to predict the future (or past) behaviour of a partial. However, due to the complexity of most arbitrary sounds, one would always find such sudden changes in the spectrum of a real sound which are totally impossible to predict using only past information. Following this idea, we split the envelope following procedure into two separate tasks. On the micro-level, we create short envelope chunks that follow some clear trend. Then, on the macro-level we merge these segments into bigger partials considering only the similarity of these chuncks.

We present the details of this procedure in the next section. Then, we discuss briefly some of the benefits, drawbacks and possible future improvements in Section 3.

## 2. ADDITIVE ANALYSIS

Additive synthesis approximates a target sound $s$ by adding together (a given number of) sinusoids:

$$s(t) \approx \sum_{i=0}^{n} A_i(t) \sin\Big(\omega_i(t)\Big), \tag{1}$$

where $A_i(t)$ and $\omega_i(t)$ are the amplitude and frequency trajectories of each partial, respectively [10]. An additive analysis method must, therefore, extract these partials from any arbitrary sound. The most convenient way of doing this consists normally of three main steps. First, one has to decompose the sound into a dataset whose elements carry meaningful information in terms of amplitude and frequency. Short-time Fourier transforms (STFT) are particularly popular for this task, although other methods – e. g. Discrete Wavelet Transform – could be adopted as well [6]. As a next step, one would filter this dataset to extract the most important data points that contribute to the sound, which we would call *tonal peaks*. Finally, one would deduce the desired trajectories using these peaks.

### 2.1. Preparatory Steps

For the first step, our analysis tool uses STFT technique with arbitrary-sized analysis windows, using zero-padding to reach an adequate (being a power of 2) FFT-window size. To find the tonal peaks, we followed the parabolic interpolation strategy described in [11]. Any alteration – e. g. due to psychoacoustical reasons – to the obtained peaks should be applied at this point[1]. At the end of this step, all tonal peaks that we found are collected into a set that we called *peak pool*.

Before continuing, let us define the absolute and relative differences of the tonal peaks $P_1(l_1, p_1)$ and $P_2(l_2, p_2)$ (here, $l_i$ is the level and $p_i$ is the pitch of $P_i$). Now we can define the points

$$P_1 - P_2|_{\text{rel}} \equiv P_1 - P_2 := \Big(l_1 - l_2, p_1 - p_2\Big) \tag{2}$$

as the relative and

$$P_1 - P_2|_{\text{abs}} := \left(\exp\frac{l_1}{l_2}, \exp\frac{p_1}{p_2}\right) \tag{3}$$

[1] These considerations will be added in a future release of KLANGPILOT.

as the absolute difference of two peaks. Following these definitions, we can also define the absolute and relative distance of the two peaks as

$$\Delta^{\varphi}(P_1,P_2) := \left( \left|(P_1-P_2)^{\mathrm{L}}_{\varphi}\right|, \left|(P_1-P_2)^{\mathrm{P}}_{\varphi}\right| \right) \qquad (4)$$

where $\varphi \in \{\mathrm{abs},\mathrm{rel}\}$, standing for 'absolute' and 'relative. It is easy to prove that both 'coordinates' of these distances are metrics.

We introduce at this point the term *virtual tonal peak* as well, being the mean of the set $\{P_1 \dots P_r\}$ of simultaneous tonal peaks according to the formula

$$\Pi := \left( \frac{1}{r}\sum_{i=1}^{r} l_i, \frac{1}{r}\sum_{i=1}^{r} p_i \right) \equiv \left( \Pi^{\mathrm{L}}, \Pi^{\mathrm{P}} \right) \qquad (5)$$

To represent the original tonal peaks that were merged into the virtual tonal peak $\Pi$ we define the variance of a virtual tonal peak as

$$\mathrm{Var}\,\Pi := \left( \frac{1}{r}\sum_{i=1}^{r} \left(l_i-\Pi^{\mathrm{L}}\right)^2, \frac{1}{r}\sum_{i=1}^{r} \left(p_i-\Pi^{\mathrm{P}}\right)^2 \right) \qquad (6)$$

Note that a normal tonal peak can actually be considered as a virtual tonal peak with zero variance, hence we don't really need to distinguish between real and virtual tonal peaks.

The most difficult part of an additive analysis method is usually the third step: that is, to build a partial tracker using the extracted tonal peaks. Our main assumption was that if a particular partial plays an important role in the sound, then it should contain at least a few loud tonal peaks. Therefore, instead of the usual way of starting the envelope matching algorithm at zero-time, our method would first search for the loudest available tonal peak and start an envelope following process based on that peak. This procedure consists of two major steps.

### 2.2. Finding Envelope Prototypes

Partial tracking is, by its nature, a paradox activity. On one hand, peaks that deviate 'too much' from a given envelope must not be taken into account. On the other hand, the process can't be too rigourous – unexpected alterations of the partial must be captured as well. To fit with these ambiguous needs, our algorithm breaks the envelope following into a micro- and a macro-level part. First, we create short-time envelope segments – called *envelope prototypes* – which follow some simple and straightforward pattern; this is the micro-level analysis. The macro-level analysis would then merge these prototypes into real envelopes.

For the microscopic analysis, we introduce a set of user-defined parameters: the maximum allowed errors for level and pitch ($\delta^{\mathrm{rel}}_{\mathrm{max}}$) as well as for amplitude and frequency ($\delta^{\mathrm{abs}}_{\mathrm{max}}$) and the so-called *time extension factor* ($\tau_{\mathrm{ext}}$), whose purpose will be explained in a moment. To create a new envelope prototype, we do the following sequence:

1. We take the loudest tonal peak from our peak pool. We also get every simultaneous tonal peak (taking into account also those which are already included in other envelope prototypes) whose distance from this loudest peak – either relative or absolute – is smaller than the user-defined errors. We merge then all these peaks into a single virtual peak as described in Equations (5) and (6). This will be the first member of our new envelope prototype.

2. We extrapolate the envelope prototype by one step forwards in time[2]. If the envelope prototype consists of a single peak, we simply duplicate that peak. Otherwise, we use cubic spline extrapolation [7].

3. We create a new virtual tonal peak using all tonal peaks whose distance from the extrapolated peak is smaller than the defined errors. If we found at least one such tonal peak, we add the result to the envelope prototype.

4. We repeat steps 2–3 in backwards-time direction as well.

5. If we found at least one new peak during steps 2–4, we repeat the process from step 2. If we didn't find any new peak, but the next time step forwards (backwards) is closer in time than the extended ending (starting) time of the envelope prototype – which is defined by the actual duration of the prototype multiplyed by $\tau_{\mathrm{ext}}$ and then added to both ends of the envelope –, we skip this time-point and jump back to step 2.

6. When the new envelope prototype is ready, we remove every tonal peak contained by the newly created prototype from the peak pool and jump back to the first step. We repeat this process until the peak pool is not empty.

There are several reasons for using cubic spline extrapolation to guess the new peaks:

- The extrapolation favourizes 'horizontal' envelopes to start with. On the other hand, even with two points it gives a linear extrapolation, taking already into account the main deviations of the partial from the very beginning of the whole process.

- Splines are smooth enough to be good candidates for partials, yet free enough to conform with unexpected changes in the trajectories.

- The model has minimal assumptions on the shape of partials compared to linear prediction systems or the classical step-by-step envelope follower processes.

[2]The time unit of the whole analysis is defined by the hop size used during the STFT.

### 2.3. Merging Envelopes

After the micro-level analysis, we would usually end up having a huge number of quite short prototypes – envelopes that normally contain no more than a few virtual or real tonal peaks. The macro-level part of the analysis consists of the creation of the final envelopes by merging the prototypes into larger blocks. This process relies on the *pseudo-cross-correlations* of the envelope prototypes.

Let $\mathscr{E}_1$ and $\mathscr{E}_2$ be two envelope prototypes. Let $\tau_{1,2}^{\min}$ and $\tau_{1,2}^{\max}$ denote the starting and ending times of these envelope prototypes, respectively (here the starting and ending times mean the *extended* times, as explained in the previous section). To get the pseudo-cross-correlation of $\mathscr{E}_1$ and $\mathscr{E}_2$ we would first find the appropriate $t^{\min}$ and $t^{\max}$ values so that $t^{\min} = \max\{\tau_1^{\min}, \tau_2^{\min}\}$ and $t^{\max} = \min\{\tau_1^{\max}, \tau_2^{\max}\}$. As a next step, we compute the following 'two-dimensional' values:

$$\forall t : t^{\min} \le t \le t^{\max} : \quad \delta_t := \delta_{\max}^{\mathrm{rel}} - \Delta^{\mathrm{rel}}\Big(\mathscr{E}_1(t), \mathscr{E}_2(t)\Big) \tag{7}$$

where $\mathscr{E}_i(t)$ is the spline-interpolated virtual tonal peak of the envelope prototype $\mathscr{E}_i$ at time $t$ (if $\mathscr{E}_i$ contains a tonal peak at the given time, $\mathscr{E}_i(t)$ would give us that tonal peak as a result, of course). Let $S$ denote the total number of $\delta_t$-s defined by Equation (7) and let $\bar{\delta}_t$ denote those $\delta_t$-s whose both 'coordinates' are non-negatives. Now we can define the pseudo-cross-correlation of $\mathscr{E}_1$ and $\mathscr{E}_2$ as

$$\mathscr{C}(\mathscr{E}_1, \mathscr{E}_2) := \frac{1}{S^2} \sum_{\forall i,j : \bar{\delta}_i \in \{\bar{\delta}_t\} \wedge \bar{\delta}_j \in \{\bar{\delta}_t\}} \frac{\bar{\delta}_i^{\mathrm{L}}}{\delta_{\max}^{\mathrm{rel,L}}} \frac{\bar{\delta}_j^{\mathrm{P}}}{\delta_{\max}^{\mathrm{rel,P}}}, \tag{8}$$

where the upper indices L and P denote the level and pitch 'coordinates', respectively.

It is easy to prove that $\mathscr{C}(\mathscr{E}_1, \mathscr{E}_2)$ is commutative and has the following properties (hence the name'pseudo-cross-correlation'):

- $0 \le \mathscr{C}(\mathscr{E}_1, \mathscr{E}_2) \le 1$.
- $\mathscr{C}(\mathscr{E}_1, \mathscr{E}_2) = 1 \Leftrightarrow \forall t : t^{\min} \le t \le t^{\max} : \mathscr{E}_1(t) = \mathscr{E}_2(t)$.
- $\mathscr{C}(\mathscr{E}_1, \mathscr{E}_2) = 0$ if and only if there is no peak in $\mathscr{E}_1$ whose relative distance from $\mathscr{E}_2$ would be smaller than $\delta_{\max}^{\mathrm{rel}}$ (and vice versa).

Based on this definition, we can get the final envelopes by computing the pseudo-cross-correlations of each pair of envelope prototypes and merging each pair where this value is above a certain user-defined limit (denoted $\varepsilon_{\mathrm{corr}}$) into a bigger envelope chunk. To merge the envelope prototypes $\mathscr{E}_1$ and $\mathscr{E}_2$, we simply take the union of all tonal peaks contained in $\mathscr{E}_1$ and $\mathscr{E}_2$. During this, we should not forget to create additional virtual tonal peaks if needed – that is, if simultaneous tonal peaks exist in the set. Note that the newly created envelope chunck is an envelope prototype as well, therefore we can go on with the envelope merging process recursively, as long as we find envelope pairs with cross-correlations exceeding $\varepsilon_{\mathrm{corr}}$.

It is important to note that we didn't assume anything about the global shape of our partials during the macro-level analysis, the only thing we used was the correlation between envelope prototypes (which is a clear measure of the level of their similarity).

## 3. POTENTIALS AND DRAWBACKS

The final set of partials produced by the method described above is adequate for the additive synthesis described by Equation (1). Nevertheless, one might notice that the variances of the virtual peaks defined by Equation (6) were not used by the analysis at all. Keeping track of these variances is, however, a promising way to extend our method. These values show us the error levels of our approximation. One could, for instance, add randomness to the obtained frequency and amplitude envelopes, where the level of the randomness would be defined by the instantaneous variance of the partials[3]. Further analysis of the variances could also lead us to find more details and patterns in the partials, like high-frequency ring modulations etc.

Another important aspect of the model is that the main analysis block won't rely on the time structure of the tonal peaks – they don't need to be arranged into the usual 'grid' that we obtain after our STFT. Thus, even if we change drastically the process that extracts the tonal peaks (for instance, by replacing the STFT with a Wavelet-like transform), our envelope follower wouldn't need to be altered.

We shall also briefly explain why we allowed during the micro-level analysis described in Section 2.2 that tonal peaks could appear in more than one envelope prototype simultaneously: in real-life situations adjacent partials usually cross each other from time to time. Envelope following methods that won't let a tonal peak to belong to several partials simultaneously have great difficulties with the treatment of such scenarios [4]. On the other hand, the complexity of the whole process won't increase too much by this 'tolerance'. To illustrate the importance of this question, we show a successfully recognised envelope cross in Figure 1.

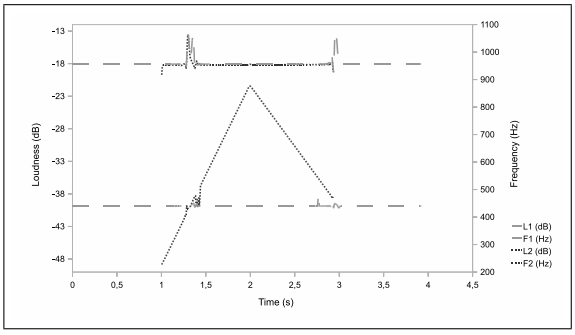


**Figure 1**. Analysis of two partials crossing each other. The analysis parameters were: $\delta_{\max}^{\mathrm{rel}} = (2\ \mathrm{dB}, 50\ ¢)$, $\delta_{\max}^{\mathrm{abs}} = (0, 5.38\ \mathrm{Hz})$, $\tau_{\mathrm{ext}} = 1$, $\varepsilon_{\mathrm{corr}} = 0.25$.

[3] This is the method that we actually implemented in KLANGPILOT.

Here, we analysed the following, simple sound:

$$s(t) := \sin(\omega t) + \begin{cases} \sin\left(\frac{3t-2}{2\text{ s}}\omega t\right), & 1\text{ s} \le t \le 2\text{ s} \\ \sin\left(\frac{4-t}{1\text{ s}}\omega t\right), & 2\text{ s} \le t \le 3\text{ s} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $\omega = 440$ Hz and $0\text{ s} \le t \le 4\text{ s}$.

After examining our results, we can see that the algorithm detects the crossing, but is very confused at the moments where these crossings happen (we can observe this on both the loudness and frequency envelopes). We assume that this is mainly caused by the use of spline extrapolation. Future research needs to address this problem, probably by refining the extrapolation method.

As a comparison, we analysed the same sound using two different methods, linear prediction [3] and the bias corrected estimation technique [9] (see Figure 2).

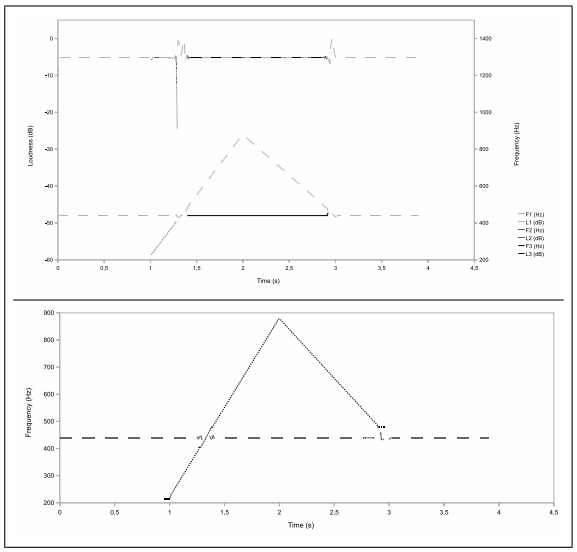


**Figure 2**. Linear prediction (upper) and bias corrected estimation (lower) analysis of $s(t)$.

As we may see, the linear prediction system failed to recognize the crossing of the envelopes, instead, it detected three different partials. On the other hand, the bias corrected estimation – after some fine-tuning of the parameters – was able to detect the crossing in a quite precise way.

## 4. CONCLUSION

Our first results with the new partial tracking process proposed in this article seem to be promising. However, there are still many unanswered questions. Particularly, a better understanding of the way how the parameters affect the final results would be essential. Also, more research should be carried out in order to determine the best practices for both the micro- and macro-level analyses. We might experiment with alternative extrapolation methods instead of splines; also the definition given in Equation (8) for the pseudo-cross-correlation could probably be refined. Nevertheless, the first results show that the basic idea of partitioning the envelope follower task into a microscopic and a macroscopic procedure will fullfill our highest expectations.

## 5. REFERENCES


[1] M. Klingbeil, "Software for spectral analysis, editing, and synthesis," in *International Computer Music Conference*, Barcelona, Spain, 2005.

[2] J. Kretz, "Extending the klangpilot score language for real-time notation," *Contemporary Music Review*, vol. 29, no. 1, pp. 29–37, 2010.

[3] M. Lagrange, S. Marchand, M. Raspaud, and m. Rault, "Enhanced partial tracking using linear prediction," in *Proceedings of the Digital Audio Effects (DAFx03) Conference*. London, United Kingdom: Queen Mary, University of London, September 2003, pp. 141–146.

[4] M. Lagrange, S. Marchand, and J.-B. Rault, "Using linear prediction to enhance the tracking of partials," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Montreal, Canada: IEEE, May 2004.

[5] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[6] *Wavelet-Based Peak Detection*, National Instruments, Jul. 2009, http://zone.ni.com/devzone/cda/tut/p/id/5432.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in FORTRAN. The art of scientific computing*, 2nd ed. Cambridge University Press, 1992.

[8] M. Raspaud, S. Marchand, and L. Girin, "A generalized polynomial and sinusoidal model for partial tracking and time stretching," in *Proceedings of the Digital Audio Effects (DAFx05) Conference*. Madrid, Spain: Universidad Politécnica de Madrid, Sep. 2005, pp. 24–29, iSBN: 84-7402-318-1.

[9] A. Röbel, "Frequency-slope estimation and its application to parameter estimation for non-stationary sinusoids," *Computer Music Journal*, vol. 32, no. 2, pp. 68–79, 2008.

[10] X. Serra, "Musical sound modeling with sinusoids plus noise," in *Musical Signal Processing*, ser. Studies on New Music Research, C. Roads, S. T. Pope, A. Picialli, and G. De Poli, Eds. Swets & Zeitlinger, 1997, pp. 91–122.

[11] X. Serra and J. Smith, "Parshl: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *International Computer Music Conference*, Champaign/Urbana, USA, 1987.