



---

# Audio Engineering Society

# Convention Paper 8477

Presented at the 131st Convention  
2011 October 20–23 New York, NY, USA

*This Convention paper was selected based on a submitted abstract and 750-word precis that have been peer reviewed by at least two qualified anonymous reviewers. The complete manuscript was not peer reviewed. This convention paper has been reproduced from the author's advance manuscript without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see [www.aes.org](http://www.aes.org). All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.*

---

## A Non-Time-Progressive Partial Tracking Algorithm for Sinusoidal Modeling

Maciej Bartkowiak<sup>1</sup>, Tomasz Żernicki<sup>2</sup>

<sup>1</sup> Poznan University of Technology, Poznań, Poland  
[mbartkow@multimedia.edu.pl](mailto:mbartkow@multimedia.edu.pl)

<sup>2</sup> Telcordia Poland Sp. z o.o., Poznań, Poland  
[tzernick@telcordia.com](mailto:tzernick@telcordia.com)

### ABSTRACT

In this paper we propose a new sinusoidal model tracking algorithm that implements a non-time-progressive way of data processing. Sinusoidal partial parameters are estimated in the consecutive frames; however, the order of establishing individual connections between partials is determined by a greedy rule within the whole signal or within a specific time window. In this way, the strongest connections may be determined early, and subsequent predictions of each trajectory evolution are based on a more reliable partial evolution history, compared to a traditional progressive scheme. As a consequence, the proposed non-progressive tracking algorithm offers a statistically significant improvement of obtained trajectories in terms of better classic pattern recognition measures.

### 1. INTRODUCTION

Sinusoidal models and hybrid sinusoidal+noise models are well-established family of signal representations for applications such as speech and audio analysis, time and pitch scaling, enhancement, restoration, source separation, automatic recognition, watermarking, compression, and synthesis. This kind of modeling usually consists of detection and estimation of sinusoidal partials in consecutive frames of signal samples, followed by partial tracking along those

frames. While the tasks of detection and estimation are well handled by fast and fairly accurate methods, the stage of tracking still remains the most challenging problem of sinusoidal modeling [1-6].

In this paper, we propose a new general strategy of tracking that may be used within the context of many tracking algorithms from a broad range that employ any form of prediction. The basic idea is to replace the usual scheme of estimating trajectories in a progressive (frame by frame) manner by a scheme that iteratively considers the best matching connections in the whole time span, and attempts to extend each existing

trajectory segment by applying two-directional prediction followed by eventual joining of shorter segments. We show that this change from time-progressive to a non-time-progressive scheme brings important benefits: the prediction of partial evolution is more reliable from the beginning, and thus less random partials are linked into trajectories.

### 1.1. Optimal tracking vs practical approach

The definition of what constitutes an optimal set of partial trajectories depends on the particular application of the model. For example, long and continuous trajectories obtained by excessive linking of partial data representing actually different sources may yield significant errors in source separation. Conversely, too fragmented trajectories resulting from too conservative connection rules are inefficient in data compression. It may be argued, that optimal tracking should result in representing actual evolutions of frequency and amplitude of each individual sinusoidal component of the original signal. However, it is an elusive goal, since the partial detection technique is usually not able to deliver sufficient data representing all individual components. Thus, every existing sinusoidal partial tracking algorithm is more or less heuristic and application specific. The often employed practical strategy is to favor smooth and continuous trajectories, because it is believed that frequencies of sinusoidal partials of music and speech signals evolve in such a way.

### 1.2. Connection criteria

Every tracking algorithm is uniquely characterized by the set of rules determining the connections between partial parameters estimated in consecutive frames. These rules include partial birth, continuation, and death criteria that may depend on direct frequency (or frequency and amplitude) difference between a currently considered endpoint of a given trajectory and partials detected in subsequent frame (or several frames) [1,2]. Usually, a connection is made for a “best match”, however the difference must not exceed a certain threshold. If there is no partial fulfilling the last requirement, a “zombie” partial is conditionally created by repeating parameters of unmatched partial from previous frame. Zombie partials help to establish connections across several frames despite of missing data, thus preventing excessive fragmentation of trajectories.

In a more sophisticated approach to tracking, the decisions regarding partial continuations are based on a wider context. This may be implemented by treating the problem of tracking as a global optimization problem of finding an undetermined number of non-coincident paths through a time-frequency (or time-frequency and amplitude) grid of partials detected in each frame. Such optimization is known to deliver satisfactory set of trajectories at the expense of high computational complexity due to the general optimization algorithms (e.g. the Viterbi algorithm) being employed to solve the problem [3]. On the other hand, global optimization still does not guarantee any long term smoothness of individual trajectories. This last requirement may be met using algorithms based on iterative application of various types of prediction [4,5,6].

In a typical prediction-based scenario, the evolution of frequency and amplitude within each already established trajectory is taken into account while considering new connections in subsequent frames. An adaptive predictor is trained on the sequence of data within the trajectory preceding current frame. It calculates the value expected in next frame which is compared to the actual data. A connection is established typically for a best matching partial, provided the difference does not exceed a given threshold (fig. 1). Otherwise, the calculated predicted values are used as provisional zombie points that help to bridge connections over a number of frames with missing data.

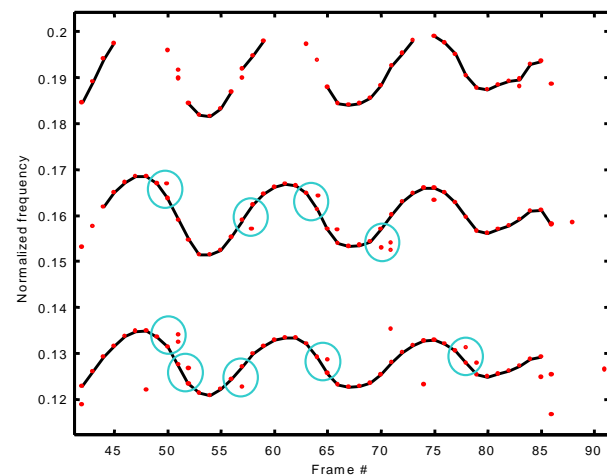


Figure 1. Prediction-based tracking example. Connections with data points are selected according to best match with predictor output, rather than minimum frequency difference (indicated by circles).

### 1.3. Connection strategy

The second key aspect of a tracking algorithm is the general strategy that determines the order in which connections are considered. The existing solutions described in the literature implement a progressive strategy, wherein connections are built frame by frame, according to the flow of time. The advantage of such approach is the possibility of online (near real-time) processing of incoming audio data. The important disadvantage is that onsets of many natural sounds are usually characterized by rapid frequency and amplitude variations which yield detection and estimation problems. A sequence of instable (and often incomplete) partial data used for training a predictor usually yields very inaccurate prediction results, and finally yields random and quite inappropriate connections, especially near the beginnings of trajectories (fig. 2).

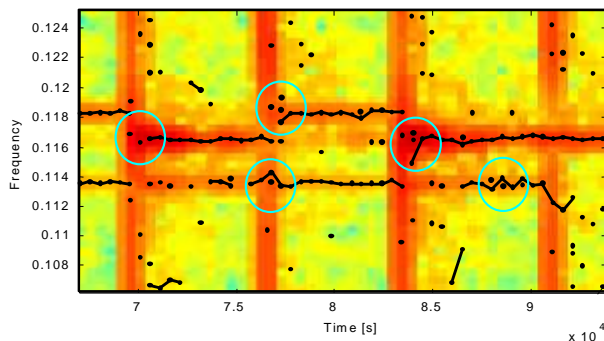


Figure 2. Illustration of tracking problems near transients due to estimation errors and inaccurate prediction results (indicated by circles).

A variant of progressive tracking that performs data processing in the time-reversed order has also been described. It is basically affected by a similar problem due to the ends of typical sounds being faded out to silence and masked by the background noise. Hence, parameter estimates of detected partials are heavily biased or contaminated by random errors, and prediction based on these values is also unreliable.

## 2. THE PROPOSED TECHNIQUE

### 2.1. The principle of non-progressive tracking

The basic idea of non-progressive tracking is to change the order of establishing connections between data points representing partials: from one determined by the order of frames to one determined by the degree of

matching. In other words, trajectories are built not from the onset of a given sound towards its end, and not from the end toward onset, but from random points in the middle (pairs of data points offering most reliable connections) and growing in all directions by joining smaller pieces into a whole.

This idea is executed by firstly seeking best “matching” connections between all possible data points in each pair of subsequent data frames, disregarding the actual order of frames. Based on these connections, prediction of further evolutions of partials is performed in both directions and subsequent data points are added iteratively.

### 2.2. The algorithm

The proposed algorithm operates on a two-dimensional data structure representing partials estimated in consecutive frames of sinusoidal analysis, as well as the connections between these data points, and the degree of matching between points in neighboring frames (score). The algorithm is iterative and requires an initialization stage.

#### 2.2.1. Initialization

During initialization, the degree of matching  $\lambda$  for all data points in all frames with all data points in subsequent frames are calculated using frequency and amplitude difference,

$$\lambda_f \{f_k^n, f_m^{n+1}\} = \min \left\{ 1 - \frac{|f_k^n - f_m^{n+1}|}{\Delta_{\max} f_k^n}, 0 \right\}, \quad (1)$$

$$\text{where } \Delta_{\max} f_k^n = \max \{ \delta_f f_k^n, \Delta_f \} \quad [\text{Hz}], \quad (2)$$

and

$$\lambda_A (A_k^n, A_m^{n+1}) = \min \left\{ 1 - \frac{|A_k^n - A_m^{n+1}|}{\Delta_{\max} (A_k^n, A_m^{n+1})}, 0 \right\}, \quad (3)$$

$$\text{where } \Delta_{\max} (A_k^n, A_m^{n+1}) = \begin{cases} \Delta^+ A & A_k^n \geq A_m^{n+1} \\ \Delta^- A & A_k^n < A_m^{n+1} \end{cases} \quad [\text{dB}]. \quad (4)$$

The above measures are normalized in the range of  $\langle 0, 1 \rangle$ , and related to predefined thresholds ( $\Delta f$ ,  $\delta f$ ,  $\Delta^+ A$ ,  $\Delta^- A$ ) that allow to control the sensitivity of the algorithm.

Note, that for the maximum frequency change  $\Delta_{\max} f$ , both absolute difference limit ( $\Delta f$ ) and relative difference limit ( $\delta f$ ) is considered (set approximately at 30Hz and 3%, respectively). This is a crucial modification w.r.t. the original MQ algorithm [1], and allows to properly cope with frequency modulation depth increasing for high-order partials of a sound spectrum, while taking into account the typical accuracy limitations of frequency estimation which is a part of sinusoidal analysis. On the other hand, for the maximum amplitude change  $\Delta_{\max} A$ , separate limits are definable for amplitude increase ( $\Delta^+ A$ ) and decrease ( $\Delta^- A$ ), typically in the order of 10dB.

After calculating the joint score  $\lambda = (\lambda_f \lambda_A)^{1/2}$  for every pair, only the one result is stored for each partial  $X_k^n$  in each frame  $n$  that gives the maximum  $\lambda_{k_m}^{n, n+1}$ . This value indicates the best matching between given partial  $k$  and the selected partial  $m$  in next frame. This same procedure is repeated for partial pairs  $X_k^n$  and  $X_m^{n-1}$  (describing the matching in opposite direction,  $\lambda_{k_m}^{n, n-1}$ ). During the main loop, the scores are updated, as soon as new connections are added between data points, which creates a context for partial prediction.

### 2.2.2. Main loop

After the table is initialized, the iterative development of actual connections starts. First of all, a sorted list of all non-zero matching scores  $\lambda_{k_m}^{n, n+1}$  and  $\lambda_{k_m}^{n, n-1}$  is created. In each iteration, a highest score indicating best matching pair is selected, and a trajectory segment between corresponding data point  $k$  in frame  $n$  and data point  $m$  in neighboring frame  $n+1$  is established (fig. 3). This connection obstructs any other possible connections between points in frame  $n+1$  and  $X_k^n$ , as well as any other possible connections between points in frame  $n$  and  $X_m^{n+1}$ . Therefore all scores in frame  $n$  that point to  $X_m^{n+1}$ , as well as all scores in frame  $n+1$  that point to  $X_k^n$  are invalidated (fig. 3), and second choice scores have to be found by recalculating the respective values of  $\lambda_{k_m}^{n, n+1}$  and  $\lambda_{k_m}^{n, n-1}$ .

The new established connection creates a new context for data points  $X_k^n$  and  $X_m^{n+1}$  or other points which may be already linked to them. Therefore, the possible connections from  $X_m^{n+1}$  to data points in subsequent frames, as well as connections to  $X_k^n$  from data points in frames preceding  $n$  are examined against existing chains. Let denote the first element of this chain as  $X_p^{n-u}$  (which may be the same as  $X_k^n$ ) and the last element as  $X_r^{n+v}$  (which may be the same as  $X_m^{n+1}$ ). The set of data

points within the chain is now a new whole trajectory that may be further extended by the use of prediction-based tracking (fig. 3,4).

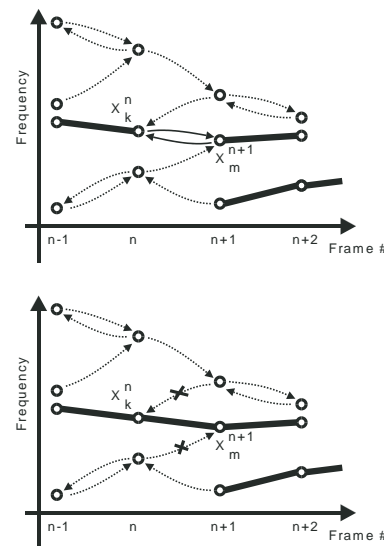


Figure 3. Connecting a best matching pair of  $X_k^n$  and  $X_m^{n+1}$  (highest scores marked by solid arrows), followed by invalidating scores marked by  $\times$ . These two scores pointing to  $X_k^n$  and  $X_m^{n+1}$  need to be re-calculated.

The corresponding values (frequency and amplitude) are temporarily collected in a data vector used for prediction,  $\mathbf{Y} = [X_p^{n-u}, \dots, X_k^n, X_m^{n+1}, \dots, X_r^{n+v}]$ . A new value  $\hat{Y}^{n+v+1}$  is calculated with the predictor actually chosen in particular application (LPC, LMS, RLS or other). These predicted values of frequency and amplitude are used in equations (1-4) instead of  $f_m^{n+1}$  and  $A_m^{n+1}$  for calculating new scores associating the data point  $X_r^{n+v}$  with a best matching data point in frame  $n+v+1$  (fig. 4). This new score replaces the existing score for data point  $X_r^{n+v}$  in the sorted list of scores, provided it is greater than zero. In case the score is zero (no data value sufficiently close to  $\hat{Y}^{n+v+1}$ ), a zombie point is inserted into frame  $n+v+1$ .

Similarly, a new value  $\hat{Y}^{n-u-1}$  is calculated from the elements of vector  $\mathbf{Y}$  in reversed order. Again, the predicted values of frequency and amplitude are used instead of  $f_m^{n+1}$  and  $A_m^{n+1}$  in calculating new scores for associating the data point  $X_p^{n-u}$  with a best matching data point in frame  $n-u-1$ . This new score replaces the existing score for data point  $X_p^{n-u}$  in the sorted list.

After the list of scores is modified and re-sorted, a new iteration follows, until the list is exhausted, which ends the algorithm.

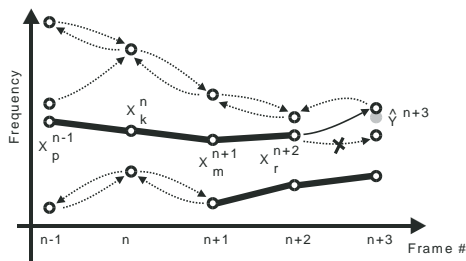


Figure 4. Calculating a new score for a trajectory endpoint  $X_r^{n+2}$ . The previous best matching (marked with  $\times$ ) is replaced by a new score (solid arrow), due to the new predicted value (grey dot).

### 2.3. Zombie points

A sequence of limited number (e.g. 2...4) of successive zombie points is allowed in each trajectory as substitutes of actual data points. Hence, the tracking algorithm is able to cope with incomplete data in several consecutive frames. Zombie points become elements of trajectories only if they provide a bridge connecting to actual data points in subsequent frames. However, if no connection is ultimately established after inserting a series of zombie points, this series is canceled, so that no trajectory ends with a zombie.

In a prediction based tracking the natural way to generate values for zombie data points is to use the predictor output ( $\hat{y}^{n+v+1}$  and  $\hat{y}^{n-u-1}$ ) in frames  $n+v+1$ , and  $n-u-1$ , respectively. Certainly, such values yield maximum matching score and would be picked up first from the sorted list of scores for creating a connection in subsequent iterations. In order to avoid catching the algorithm into an infinite loop of generating successive zombies, a dummy score value is used instead. Setting this value close to zero allows for penalizing the usage of zombie points.

### 2.4. Implementation issues

The proposed algorithm has been implemented in the Matlab environment extending the SinMod toolbox which is an enhanced implementation of sinusoidal and hybrid sinusoidal model and codec available online [7]. The data structure is implemented as a matrix of *struct* type elements, and integer indices are used as pointers. There is a configuration data structure containing all the

necessary parameters as fields. The resulting sinusoidal trajectories are returned in a form of sparse two-dimensional matrices. Every sinusoidal parameter (frequency, amplitude, phase) is stored in a separate matrix, wherein every trajectory is represented as a separate row with non-zero values within a range of columns indicating an active state for a corresponding selection of audio frames.

The presented algorithm is certainly more time demanding compared to a simple progressive one: in the current implementation the operation speed is approximately 10x slower. This is partially a result of the necessity to re-calculate the affected scores each time a connection between data points is established. This disadvantage has been deliberately accepted for the sake of limited memory resources in Matlab environment.

## 3. EXPERIMENTAL RESULTS

### 3.1. Objective evaluation methodology

Objective assessment of the quality of tracking is a challenging problem, because there is no clear definition of perfect tracking. Furthermore, the difference between a good tracking algorithm and an even better one is usually not spectacular and may be observed in very particular conditions only.

It has been earlier proposed to use the energy of the residual signal from sinusoidal model to assess the accuracy of modeling on which tracking accuracy has significant impact [8]. However, the residual energy also significantly depends on the accuracy of partial detection, estimation, and synthesis. Thus, the result of such evaluation is heavily biased by these factors.

Instead, we propose to assess the tracking algorithm using a common pattern recognition methodology. A set of artificially generated trajectories (the ground truth, GT) is prepared, from which the raw data (without connections between data points) is extracted and fed to the tracking algorithm under test. The tracking results are compared to the GT using every pair of actually connected data points. The figures of true positives (TP), false positives (FP) and false negatives (FN) are thus easily determined. Similarly to the receiver operator characteristic (ROC) curves, the values of true positives rate and false positives rate depend on algorithm settings, such as  $\Delta f$ ,  $\delta f$ ,  $\Delta^+A$ , and  $\Delta^-A$ . Since this dependence is quite nonlinear in practice, we show

sets of points rather than regular ROC curves. This comparison does not take into account the importance of connections.

### 3.2. Experiments with synthetic data

The test data consisted of artificially generated sinusoidal parameters that would be perfectly detected and estimated for a hypothetical signal containing a mixture of two modulated harmonic sounds (fig. 5). The two corresponding harmonic series were frequency and amplitude modulated and the artificially created trajectories served as the GT in our experiment.

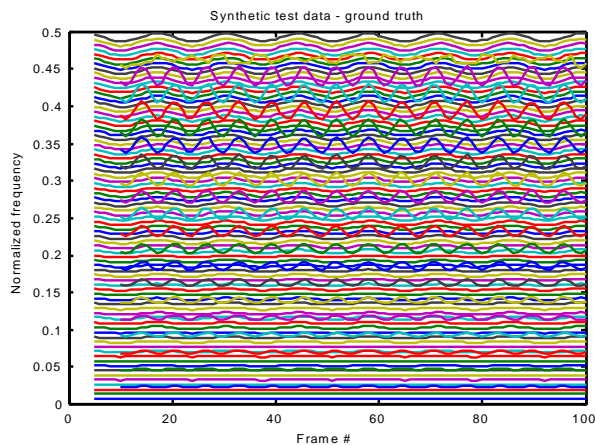


Figure 5. Synthetic test data (GT reference) used in the experiment

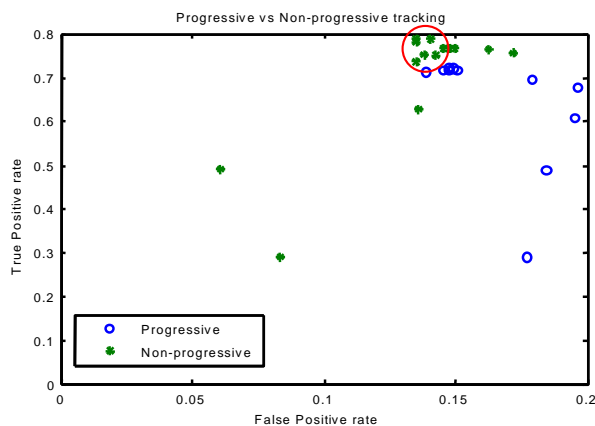


Figure 6. ROC-like performance of the non-progressive versus progressive tracking obtained for a range of parameter settings

The idea for the experiment was to observe the performance of the non-progressive algorithm with respect to a progressive algorithm that employed identical tracking criteria and same prediction scheme. For this purpose, we varied the scoring parameters ( $\Delta f$ ,  $\delta f$ ,  $\Delta^+A$ , and  $\Delta^-A$ ) in a wide range and performed tracking on the synthetic data set.

As it may be observed in fig 6, for most settings the non-progressive tracking outperforms the progressive tracking algorithm by producing slightly less invalid connections and slightly more proper connections. The region of best performance (circled) also exhibits a small advantage of non-progressive tracking.

### 3.3. Experiments with natural data

The algorithm has been tested on a range of natural audio signals including single instrument sounds and music of various styles. In the following figures we illustrate three types of benefits of non-progressive tracking. For this purpose, we show in detail three spectrograms with trajectories overprinted in black (the progressive tracking) and white (non-progressive tracking). Zombie data points generated by each algorithm are marked by circles. The trajectories have been artificially shifted in frequency for the sake of picture clarity.

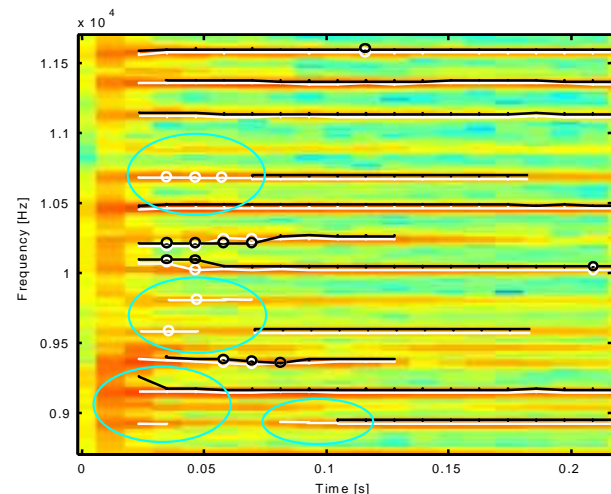


Figure 7. Difference between progressive (black) and non-progressive (white) tracking results for an audio excerpt of a harpsichord solo.

For a sound featuring sharp and wideband transients, an excessive number of partials is detected in frames

corresponding to note onset. The progressive algorithm exhibits a tendency to capture random data points and to seek continuation in points representing actual sinusoidal partials in subsequent frame. This leads to instable beginnings of affected trajectories. The non-progressive algorithm does not start a trajectory near transient, but in the middle, developing its connections in both directions, thus making more reliable decisions (fig. 7).

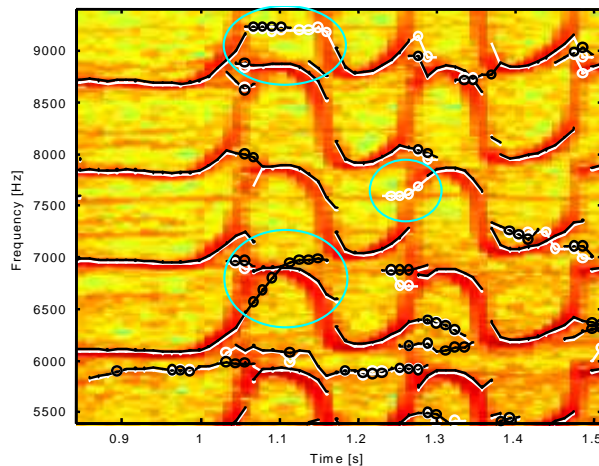


Figure 8. Difference between progressive (black) and non-progressive (white) tracking results for an music excerpt featuring brass ensemble.

For a sound featuring deep frequency and amplitude modulations sometimes it is not possible to detect all partials with rapidly varying parameters. In these situations a prediction based tracking algorithm produces a considerably high amount of zombie data points. The progressive algorithm tends to overshoot on modulated segments (fig. 8) while non-progressive one performs a more conservative bridging of broken segments.

A typical progressive tracking problem is shown in fig. 9. An initial unfortunate connection of random points starts a heavily slanted trajectory featuring a series of zombie points and finally connecting to a completely random data point. All this results in an audible chirp artifact. At the same time, non-progressive algorithm yields a more reliable trajectory that follows an actual sinusoidal component, because the sequence of zombie points is obtained by joining segments developing in opposite directions.

In general, it may be observed, that for certain configurations of data points representing sinusoidal

partials, the non-progressive tracking offers advantages in terms of producing less random and more reliable connections.

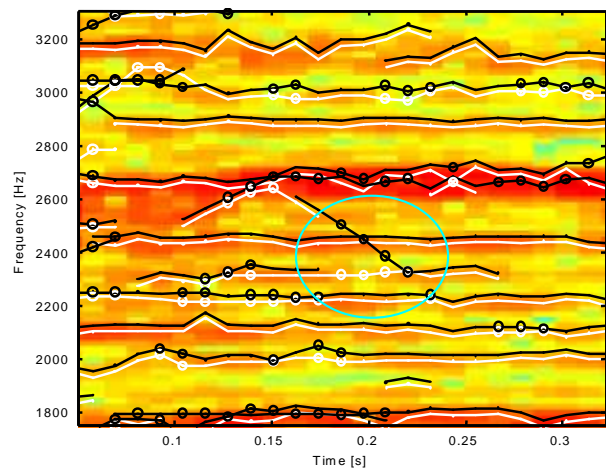


Figure 9. Difference between progressive (black) and non-progressive (white) tracking results for a complex music excerpt.

#### 4. CONCLUSIONS

A new approach for sinusoidal partial tracking has been proposed in this paper. We have shown results for synthetic as well as natural signals based on the implementation in Matlab. We have tested a number of strategies and configuration options to demonstrate differences between the non-progressive and progressive tracking. The main advantage of the proposed technique is more reliable approximation of sinusoidal trajectories in comparison to progressive tracking algorithms. Especially, we have achieved good results for sounds rich with deep frequency and amplitude modulations. However, this benefit is occupied by a significantly increased computational complexity. The algorithm cannot be applied in online/realtime processing.

#### 5. ACKNOWLEDGEMENTS

This work was supported by publish funds for scientific research under a research grant of Polish Ministry of Science and Higher Education no. N517 394838.

## 6. REFERENCES

- [1] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 4, pp. 744–754, 1986.
- [2] J. Smith and X. Serra, "PARSHL: A program for the analysis/synthesis of inharmonic sounds based on a sinusoidal representation," *International Computer Music Conference*, 1987.
- [3] P. Depalle, G. Garcia, and X. Rodet, "Tracking of partials for additive sound synthesis using hidden Markov models," *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 242–245, 1993.
- [4] M. Lagrange, S. Marchand, M. Raspaud, and J. Rault, "Enhanced Partial Tracking Using Linear Prediction," *Proceedings of the Digital Audio Effects (DAFx03) Conference*, vol. 141, 2003, p. 146.
- [5] M. Lagrange, S. Marchand, and J.-B. Rault, "Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1625–1634, July 2007.
- [6] Nunes, L.D.O.; Merched, R.; Biscainho, L.W.P. "Recursive Least-Squares Estimation of the Evolution of Partial in Sinusoidal Analysis," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.
- [7] M. Bartkowiak, T. Żernicki, "SinMod - audio sinusoidal modeling toolbox for Matlab", available at [http://www.multimedia.edu.pl/audio\\_research](http://www.multimedia.edu.pl/audio_research)
- [8] M. Lagrange, S. Marchand, "Assessing the Quality of the Extraction and Tracking of Sinusoidal Components: Towards an Evaluation Methodology", *Proceedings of the Digital Audio Effects (DAFx06) Conference*, Canada, 2006.
- [9] T. Fawcett, "An introduction to ROC analysis", *Pattern Recognition Letters*, vol. 27, no. 8, 2006.