Adaptive oscillator networks for partial tracking
and piano music transcription

Matija Marolt
Faculty of Computer and Information Science
University of Ljubljana, Tržaška 25
1000 Ljubljana, Slovenia
matija.marolt@fri.uni-lj.si

Abstract
This paper presents our recent work in developing a system for transcription of polyphonic piano music. Our goal is to build a system that would automatically transcribe polyphonic piano music from the audio signal, transcribing note onsets and offsets. The system consists of three main stages: filtering, partial tracking and note extraction. The paper presents our partial tracking method based on adaptive oscillator networks. These are used to extract partial tracks of piano notes from the time-frequency transformed audio signal. Extracted partial tracks and amplitude envelopes are then used by neural networks in the note extraction stage to perform the transcription.

1. Introduction
Music transcription (polyphonic pitch tracking) is a difficult cognitive task not just for current computer systems, but also for trained humans. Separating notes from a mixture of other sounds, which may include other notes played by the same or different instruments or simply background noise requires robust algorithms with performance that should not deteriorate too much when noise increases.

Many current systems use some kind of a time-frequency transform and peak-picking algorithm to extract partial tracks from the audio signal and then use statistical methods to group these tracks into notes. Some systems (i.e. [Rossi, 1998]) first calculate sound source models of instruments and then try to transcribe music performed by these same instruments. Others use no such models.

We were surprised, that most systems do not employ any kind of machine learning algorithms, which have proved to be successful in other pattern recognition domains (speech recognition, ...). Therefore, our first goal was to assess the performance of several standard feed forward neural network models for transcription. We have limited ourselves to transcription of polyphonic piano music, thereby making the piano the only instrument in our experiments. Results obtained with such a very simple system can be found in [Marolt, 1999].

We have extended our system by adding a partial tracking stage performed by networks of coupled adaptive oscillators. As can be seen from the results, this improves transcription accuracy and leads to a more robust system.

2. The System
Our current transcription system attempts to correctly determine notes and their starting times in a polyphonic piano performance. The system is composed of three main stages, depicted in figure 1 and described below.

Fig. 1. Structure of the system

2.1 Filtering
The first stage of our system transforms an audio signal of a piano performance into time-frequency space. The transformation is done by a commonly used auditory model emulating the functionality of human cochlea. The model first uses a gammatone filterbank to split the signal into several frequency channels. The filterbank consists of an array of bandpass filters with near constant-Q bandwidth in middle and high frequencies. Center frequencies of filters lie between 50 and 10000 Hz. We use the filter implementation based on [Patterson, et. al, 1992] and implemented by [Slaney, 1993].

Subsequently, the output of each gammatone filter is processed by the Meddis’ model of hair cell transduction [Meddis, 1986]. The hair cell model
converts each gammatone filter output into a probabilistic representation of firing activity in the auditory nerve, incorporating well-known effects such as saturation and adaptation.

2.2 Partial Tracking

Output of the filtering stage is a set of frequency channels containing quasi-periodic firing activities of inner hair cells. Many systems that employ such an auditory model compute a so-called correlogram to extract periodicity information from each channel [Slaney and Lyon, 1990]. A correlogram is formed by computing an autocorrelation in each frequency channel. The result is a frame-by-frame representation of the audio signal, where each frame in time contains a two dimensional representation with channel center frequency and autocorrelation lag represented on orthogonal axes. The autocorrelation lag with maximum value in each channel gives an estimate of maximum periodicity in that channel at a given time. A summary autocorrelation (summed across frequency channels) can then be computed to give a total estimate of periodicity at a given time. Recently, [Martin, 1999] has used running autocorrelation to calculate a continuous correlogram with logarithmic lag spacing.

We chose a different path for extracting periodicity information from the filtered sound. The final output of our model is a set of estimated strengths of groups of partials with center frequencies equal to frequencies of piano notes.

Our model is based on networks of coupled adaptive oscillators. Adaptive oscillators are a class of oscillators, which adapt their phase and frequency in response to external input. When an adaptive oscillator is presented with a periodic input signal, it tries to sync to the signal by adjusting its phase and period to that of the input signal. By observing the frequency of a synced oscillator, we can make a more accurate estimate of the frequency of the driving input signal. We are using a simplified version of the Large-Kolen adaptive oscillator model [Large and Kolen, 1994] in our experiments.

An oscillator has three variables that change with time: phase, period, and output. Phase is defined as

\[
\phi(t) = \frac{(t - t_o)}{p} \quad (1)
\]

where \( t \) is time, \( p \) the period of oscillation, and \( t_o \) the time at which the oscillator expects an event to occur. When \( t \) reaches \( t_o \), phase becomes zero and the oscillator fires (the internal activation of the oscillator reaches maximum). When a periodic stimulus \( s(t) \) is presented to the oscillator, it tries to adjust its phase and period to that of the input stimulus, so that it fires in sync with the stimulus. Phase and period are updated according to the following formulas:

\[
\Delta s = \eta_1 s(t) \frac{D - \mathrm{sech}^2 \chi}{2\pi} (\cos 2\pi p(t) - 1) \sin 2\pi \phi(t) \quad (2)
\]

\[
\Delta p = \eta_2 s(t) \frac{D - \mathrm{sech}^2 \chi}{2\pi} (\cos 2\pi p(t) - 1) \sin 2\pi \phi(t)
\]

\( \eta_1 \) and \( \eta_2 \) are parameters that determine the strength of synchronization to the stimulus, \( \chi \) is a fixed parameter defining the receptive field of the oscillator (the original Large-Kolen oscillator also adjusts this parameter, we decided to keep it constant). The output of our oscillator indicates how successfully it synced to its driving signal; the higher the value, the better the sync.

Each oscillator also has its so-called preferred frequency (or period). This is the oscillator’s initial frequency and an oscillator is only allowed to sync to frequencies that are up to one semitone higher or lower than its preferred frequency. This prevents oscillators to drift away and sync to arbitrary input frequencies.

In our model, we feed each output channel of the filtering stage (each output of the Meddis’ hair cell model) to an input of an adaptive oscillator; a channel with center frequency \( f \) will be connected to an oscillator with preferred frequency \( f \). When an audio signal containing components at \( f \) is passed through the filtering stage, the output of channel \( f \) is quasi-periodic and the oscillator connected to that output synchronizes to it, producing a high output value. Because oscillators are adaptive, they can also adapt to slight changes in the frequency of their driving signal and stay in sync in cases of small frequency modulations or beating in the input signal. Each synced oscillator therefore represents and follows a partial track.

We took the model one step further and coupled harmonically related oscillators together to form oscillator networks representing groups of partials. Each network consists of up to ten oscillators. The preferred frequency of the first oscillator in the network (base frequency of the network) is tuned to frequency of one of 88 piano notes (A1-C9 MIDI notation). Preferred frequencies of other oscillators in the network are integer multiples of the preferred frequency of the first oscillator (see figure 2). Each oscillator in a network is internally coupled to all other oscillators.

\[
\text{Fig. 2. A network of coupled oscillators estimating the strength of a group of partials at frequencies } f, 2f, 3f, ...
\]

\[
\text{Frequency and strength of the partial group}
\]

\[
\text{Meldis’ HC output channels centered at } f, 2f, 3f, ...
\]
An oscillator network has two output variables:

- **strength of the partial group**, which is calculated as a weighted sum of outputs of all network oscillators. The outputs are weighted, so that oscillators with lower preferred frequencies (first few partials) have more impact on the group’s strength than higher partials.

- **average partial group frequency**, which is calculated as the weighted average of frequencies of synced oscillators.

The entire network functions as follows:

- each oscillator with preferred frequency $f_i$ tries to sync to its input coming from the channel with center frequency $f_i$ and updates its phase, frequency and output accordingly.

- each synced oscillator updates the frequencies and outputs of all other oscillators in the network. Frequencies are updated to approach the average partial group frequency, while outputs are updated according to the following formula:

$$o_j = o_j + \omega_i \left( o_i e^{-1000(f_i - f_j)^2} \right)$$

where $o_i$ and $o_j$ are outputs of respective source and destination oscillators, $f_i$ and $f_j$ their current frequencies and $\omega_i$ the weight between the two oscillators. If frequencies of the two oscillators are sufficiently close to each other (the exponential gaussian factor), the output of the destination oscillator grows in proportion to the weight and output of the source oscillator.

Final outputs of all oscillator networks are 88 channels, each containing the strength of a partial group and the average frequency of that group. An example of such output, obtained from the transcription of Bach’s French Suite No. 1, as well as the excerpt’s amplitude envelopes and their combination can be seen in figure 3; time is represented on horizontal axis, frequency on vertical.

### 2.3 Onset Detection

We also implemented a simple onset detector to detect note onsets from the audio signal. First, we divided gammatone filterbank output channels centered at piano note frequencies into eleven overlapping groups, each containing 12 channels covering an entire octave. Within each group, channels are full-wave rectified, smoothed with a one-pole filter with 5 ms time constant and averaged together.

Peaks and their attack strengths are then determined for each group and peaks with attacks stronger than 3 dB are retained. If such peaks appear in four or more channel groups within a 50 ms time frame, an onset is detected. Its time and strength are calculated as the centroid of all peaks in all groups within the time frame.

The algorithm works quite good for piano performances, detecting around 99% of all onsets and reporting few extra onsets not present in the input.
for each network included approx. 30000 chords with 1/3 of chords containing the target note.

Outputs of all 88 neural networks represent the occurrence of piano notes in the input audio signal. Outputs are a time series of numbers, indicating how each network classified its input at a certain point in time (a high output value means a note is present, otherwise not). To obtain the final list of notes, we perform simple time averaging in each output channel to reduce noise and prevent that isolated high network activations cause a note detection. Several consecutive high activations are needed for a note to be present in the final output. The end result of transcription is a list of notes and their starting (and ending) times.

3. Results

We tested the system by transcribing several solo piano performances. In order to evaluate the results easier, we obtained the performances by rendering some MIDI files with different piano samples. The pieces ranged from very simple Bach’s Two-part Inventions to more complex excerpts from Tchaikovsky’s Nutcracker Suite.

Transcription results obtained are given in table 1. Results are given for transcriptions of five pieces: J.S. Bach’s Contrapunctus 12 from Art of the Fugue, Partita No. 1 (BWV825), French Suite No. 1 (BWV812), Tchaikovsky’s Nutcracker Suite Miniature Overture and Waltz of the Flowers. The second and third columns of table 1 represent the average and maximum polyphony of transcribed pieces. The fourth column (notes found) represents the percentage of notes in each piece that were correctly transcribed. The fifth column (extra notes) represents the number of additional notes that were found, but were not present in the input.

<table>
<thead>
<tr>
<th>piano piece</th>
<th>avg. poly</th>
<th>max. poly</th>
<th>notes found</th>
<th>extra notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrap. 12</td>
<td>1.8</td>
<td>5</td>
<td>95%</td>
<td>13%</td>
</tr>
<tr>
<td>Partita No. 1</td>
<td>2.6</td>
<td>6</td>
<td>94%</td>
<td>15%</td>
</tr>
<tr>
<td>French Suite</td>
<td>3</td>
<td>6</td>
<td>91%</td>
<td>14%</td>
</tr>
<tr>
<td>Nutcr. ovr.</td>
<td>3.1</td>
<td>6</td>
<td>90%</td>
<td>15%</td>
</tr>
<tr>
<td>Nutcr. waltz</td>
<td>5</td>
<td>15</td>
<td>81%</td>
<td>25%</td>
</tr>
</tbody>
</table>

Table 1. Transcription results

Most of the errors (either missing notes or extra notes) - over 50% - are octave or similar errors. The other most common source of errors are very short notes (less than 100 ms) or notes played very quickly one after another. When compared with standard TF methods we used previously, oscillator networks reduce the dimensionality of neural networks’ input space and improve transcription accuracy. The number of extra notes has been reduced and most of the errors are of harmonic nature and are therefore not as critical as before. Our future efforts will be directed towards reducing these types of errors.

4. Summary and Future Work

Partial tracking with oscillator networks has improved results from our previous experiments, but there is still a lot of room for improvements:

- feedback connections should be added to the system. They could be used to suppress notes in the input signal that were already found by the system and thereby reduce the number of octave and other harmonic errors.
- in the current system, networks are trained on a different domain (chords) than they are used on (transcription). We plan to add a retraining stage in which all of the networks will be retrained within the context of the system on polyphonic piano performances.
- the postprocessing stage could be improved by take into consideration not only neural networks’ outputs and onset detection, but also other factors, such detected partials, amplitude envelopes...