E85.2607: Lecture 9 – Sinusoidal Modeling
Similar to phase vocoder, but assumes that signal is sum of sinusoids with smoothly varying parameters:

\[ x[n] \approx \tilde{x}[n] = \sum_k a_k[n] \cos(\omega_k[n]) \]

- Freq. domain representation analogous to Fourier series
- Flexible representation for transformations...
SMS overview

Analysis

Extract sinusoidal tracks from STFT

Residual contains portion of signal that is not well modeled by sinusoids

Synthesis

Each track controls an oscillator
Analysis overview

1. Break signal into frame ala STFT
   - Be aware of time-frequency tradeoff

2. Pick peaks in each frame
   - Often expect to find peaks in harmonic series due to common pitch
   - Search for common factor

3. Organize spectral peaks into time-varying sinusoidal tracks
   - Expect sinusoids to drift in amplitude and frequency
Review: Time-frequency tradeoff

\[ X[k, m] = \sum_{n=0}^{N-1} x[n + mL]w[n]e^{-j\frac{2\pi kn}{N}} \]

DFT length \( N \)

- Window determines freq. resolution
- Must be long enough to resolve harmonics
  \( \sim 2 \times \) longest pitch period (\( \sim 50 - 100 \) ms)
- Too long \( \rightarrow \) blurred \( a_k[n] \)

Hop size \( L \)

- typically \( N/2 \) or \( N/4 \)
- small hop \( \rightarrow \) simpler interpolation over time

- good freq. resolution
- bad time resolution
- bad freq. resolution
- good time resolution
Peak picking

Find local maxima in each DFT frame

Accuracy of peak detection is limited to half a sample
  - Can zero pad, but can be expensive

Alternative: Frequency resolution smaller than one bin using quadratic interpolation
Not all peaks correspond to a stable sinusoid

- Only retain peaks larger than some threshold
- Only retain peaks consistent with harmonic series of underlying pitch
  - need pitch tracker, input must be monophonic
  - pitch-synchronous analysis?
- Look for stable parameters in adjacent frames
  - e.g. stable phase derivative in time and frequency
Peak continuation

- Connect peaks in adjacent frames to track sinusoid trajectory
- Lots of different approaches
  - e.g. Macaulay-Quatieri approach: Greedily attach peak in current frame to closest peak in next frame
- Ambiguous if large frequency changes
Track formation heuristics

- Connect peaks in adjacent frames to form sinusoids. Can be ambiguous if large frequency changes.
- Unclaimed peak → create new track.
- No continuation of track → termination.

Lots of other potential rules:
- Min track length to avoid spurious peaks
- Max allowable frequency deviation between adjacent frames
- Max silent gap length
- Max number of tracks per frame

Tricky to implement . . .
Each sinusoid track \( \{a_k[n], \omega_k[n]\} \) drives an oscillator.

Interpolate parameters to avoid clicks at frame boundaries.

Faster method: synthesize DFT frames, then overlap-add.
Sinusoid + noise model

- Sinusoids is not a good fit for all types of signals (e.g. noise)
- Sometimes want to retain residual in addition to sinusoids

\[ x[n] = \sum_{k} a_k[n] \cos(\omega_k[n] n) + e[n] \]

- Model residual as white noise passed through time-varying filter
Sinusoidal subtraction

original sound $x(n)$

synthesized sound with phase matching $s(n)$

residual sound $e(n) = w(n) x(x(n) - s(n))$, $n = 0, 1, \ldots, N - 1$
Modeling the residual

- Many options for modeling residual filter parameters
- Can use LPC to approximate shape of residual
- or simply smooth magnitude spectrum ala channel vocoder
Residual synthesis

- Spectral magnitude approximation of residual
- Random spectral phase
- Synthesized sound
- Synthesized sound with window
Synthesis: Putting it all together

- Sine frequencies
- Sine magnitudes
- Sine phases
- Spectral sine generation
- Magnitude spectrum
- Phase spectrum
- Polar to rectangular conversion
- Window generation
- Window
- Complex spectrum
- IFFT
- Output sound
- Residual spectral data
- Spectral residual generation
- Magnitude spectrum
- Phase spectrum
- Polar to rectangular conversion
Sin + noise - example

- Removing sines reveals noise & transients
- Different representation approaches...

Guitar - original

Guitar - sinusoid reconstruction

Guitar - residual (original - sines)
### Examples

<table>
<thead>
<tr>
<th></th>
<th>Orig</th>
<th>Sin</th>
<th>Residual</th>
<th>SMS</th>
<th>Transformed</th>
</tr>
</thead>
<tbody>
<tr>
<td>flute</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>clarinet</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>guitar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>drums</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Applications

Loads of applications

- Similar to other TF analysis-synthesis techniques
- but parameters more convenient for some transformations
  - can treat spectral shape independent from harmonics
  - can treat different tracks independently
- Filtering with arbitrary time resolution

Timescale modification

- Sinusoidal description very easy to modify e.g. changing time base of sample points
- Frequency stretch preserve formant envelope?
Applications: Frequency transformations

- Partial-dependent frequency scaling
  
  ![Graph showing partial-dependent frequency scaling](image)

  e.g. pseudo inharmonicities of higher partials in piano sound

- Frequency stretching: \( \hat{\omega}_k[n] = p \ast \omega_k[n] \)
  
  ![Graph showing frequency stretching](image)

  e.g. pitch-shift without preserving timbre

- Spectral shape shift
  
  ![Graph showing spectral shape shift](image)

- Quantize pitch (autotune)
Still more applications

- Effects we’ve seen before:
  - **Vibrato** modulate $\omega_k[n]$ with LFO
  - **Tremolo** modulate $a_k[n]$ with LFO
- Hoarseness: boost residual relative to sinusoidal components
- Morphing between sounds
- Gender change: shift pitch and formants separately
- Singing voice synthesis/conversion (**Vocaloid**) 🎵
- Separate transients from steady-state harmonics 🎵 🎵
- ...

E85.2607: Lecture 9 – Sinusoidal Modeling
Tools: SPEAR

From Michael Klingbeil: http://www.klingbeil.com/spear/
Bringing it all together

**PVOC**  \( X[k, n] = A[k, n] \ e^{j\omega[k,n]} \)
- Magnitude and phase of fixed oscillators

**Source-filter**  \( X[k, n] = E[k, n] \ H[k, n] \)
- Filter captures spectral shape (smoothed \( A[k, n] \))
- Source is whatever is left - typically pulse train corresponding to harmonic series

**Sinusoids**  \( X[k, n] = A_k[n] \cos(\omega_k[n]) + E[k, n] \)
- Organize input into oscillators with time-varying frequency (harmonic tracks)
- Sinusoid magnitudes \( A_k[n] \) approximate spectral shape
- Frequencies \( \omega_k[n] \) encode source information
DAFX  Chapter 10 - Spectral Processing