Engineering 100
Music Signal Processing
Professor Andrew E. Yagle

- Computer synthesis of musical signals using:
- Digital turntable synthesis of instruments by upsampling/downsampling of one snippet;
- Additive synthesis of artificial instruments: Make your own kind of music.

Computer Synthesis of Music

- Compose your own music using existing instruments (need a snippet [short sample]): Digital turntable: Up/down-sample snippet.
- Make your own kind of music using artificial computer-generated instruments! Additive synthesis: Sum Fourier series.
- Subtractive and FM synthesis not covered.

Analog Turntable Synthesis

- Take a recording of (say) single clarinet note.
- Speed up/slow down tape to change its pitch.
- Only need variable-speed motor or turntable.
- Alvin & the Chipmunks: Double tape speed.
- Cut and paste (literally!) or dub tape→music.

Digital Turntable Synthesis

- Take a recording of (say) single guitar note.
- Sample (digitize)→string of numbers→note.
- Easy to cut to desired duration & concatenate different notes & add instruments in Matlab.

- But how do we alter pitch if have one note?
- Upsample & downsample & Circle of Fifths.

Doubling/Tripling Pitch

- Double pitch of the musical note stored in X: >>Z=[X X];Y=Z(1:2:length(Z));
- Take every other sample→halve sampling rate. Take X twice so duration of Z=duration of X. All sinusoids are now oscillating twice as fast. Make sure X bandlimited to ¼(sampling rate).
- Triple pitch of the musical note stored in X: >>Z=[X X X];Y=Z(1:3:length(Z));

Halving Pitch

- Insert a zero between each pair of samples x[n]: {…0,x[-2],0,x[-1],0,x[0],0,x[1],0,x[2],0…}
- Lowpass filter (half-band) this upsampled x[n].
- Result has pitch halved, but duration doubled.
- Why? See synth.pdf on the CTools website.
- >>L=length(X);Z=[X;zeros(1,L)];F=fft(Z(:)); >>F(L/2+1:3*L/2+1)=zeros(1:L+1);Y=ifft(F);

Altering Pitch of a Snippet

- Since all semitones are related by ratios of small integers, can obtain all semitones from any single one by up/down-sampling the one.
- "Upsampling and downsampling" is also known as "interpolation and decimation."
- Another way: Use the musical Circle of Fifths

Musical Circle of Fifths

$\log_{1.5}$[f /440 Hz]	0	1	2	3	4	5	6	7	8	9	10	11
octave	0	0	1	1	2	2	3	4	4	5	5	6
Hertz	440 440	660 659	495 494	742 740	557 554	835 830	626 622	470 466	705 698	529 523	793 784	595 587
Note	A	E	B	F#	C#	G#	D#	A#	F	C	G	D

Repeatedly downsampling by 2 and upsampling by 3 (so no aliasing)→ can obtain all 12 semitones from any one. Why? $1.5 \approx 2^{7/12} \rightarrow (1.5)^{12} \approx 2^7$.

Musical Circle of Fifths

- Place 12 semitones in order on a clock face.
- Take 12 jumps of 7 hours each around face.
- Land on all 12 semitones (relatively prime).
- Jump of 7 semitones=5 whole tones="fifth" =frequency ratio of 1.5 (A→E: 440→659).
- Worked perfectly until "Well-Tempered Clavichord" (Bach); still works pretty well.

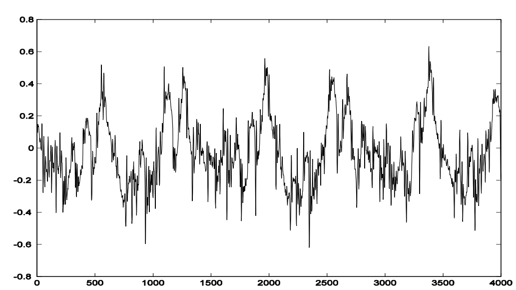
Using Circle of Fifths for Synthesis

- Start with snippet of instrument at G (392 Hz)
- Downsample Y=X(1:2:length(X)) for 784 Hz
- 12 times: Downsample by 2 & upsample by 3
- 12 times: F=fft(X);Y=(3/2)*real(ifft([F(1:L/2) zeros(1,L/2+1) F(L/2+2:L)]));Y=Y(1:L);
- No aliasing: frequencies down to 2/3 each time
- Have all 12 semitones! Get correct durations.

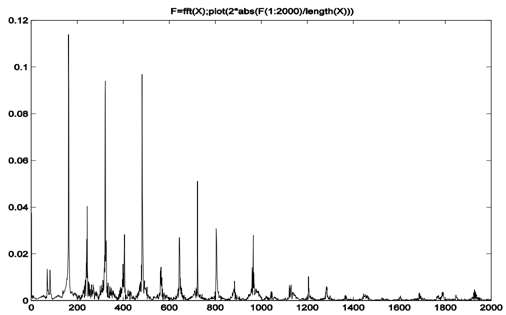
Example: Guitar Note Snippet

- >>[X,FS]=wavread('guitar.wav');
- >>length(X)→114000 and >>FS→44100.
- Duration of X=114000/44100=2.585 sec.
- Period of note X=1425/44100=.0323 sec.
- "Pitch" of note X=44100/1425=30.9 Hertz.
- Fund=(81-1)(44100)/(114000)=30.9 Hertz.
- 3 instruments: each length $32768=2^{15}$ (why?)

Guitar Note: Time Waveform



Guitar Note: Line Spectrum



Additive Synthesis of Music

- Musical note "A" signal is periodic. Expand:
- $x(t)=c_1\cos(2\pi 440t)+c_2\cos(2\pi 880t)+\ldots$
- Amplitudes c_k determine timbre of sound.
- Omit phases, since we can't perceive them.
- Finite #terms, since harmonics can't be heard if > 10 kHz. Also: < ½(sample frequency).

Additive Synthesis of Music

- c_k are specified by the instrument. BUT:
- Design your own instrument: Choose c_k!
- Hammond and pipe organs: do physically: Opening organ pipe stops varies the c_k.
- See synth.pdf for two examples of c_k sets.
- Only problem: Generating and summing the Fourier series: Lot of real-time computation.

Fast Generation of Sinusoids

- No need to use Matlab to generate sinusoids.
- Recall: $x[n]=A\cos(2\pi Fn+\theta)$ satisfies formula
- $x[n+1]+x[n-1]=2\cos(2\pi F)x[n]$. Rearrange to:
- $x[n+1]= 2\cos(2\pi F)x[n]-x[n-1]$. Initial conditions:
- $x[0]=A\cos(\theta)$. $x[1]=A\cos(2\pi F+\theta)$. Recursively.
- Each recursion: One addition and multiplication.
- Do in parallel in Matlab-Will save computation.

Subtractive Synthesis of Music

- Instead of generating harmonics and summing.
- Start with harmonics from a simple waveform (square wave, triangle wave, rectified sine) that is easy to generate with analog circuits.
- Filter waveform with analog lowpass filter.
- Result has different harmonics and timbre.
- Used in 1970s (Moog synthesizer).

FM Synthesis of Music

- FM=Frequency Modulation (vary f with t).
- $x(t)=\cos(2\pi[f_0+I\cos(2\pi f_1 t)]t)$. Can add θ's.
- Frequency varies sinusoidally around f_0 Hertz. This frequency varies with frequency f_1 Hertz. Amplitude of variation=I=modulation index.
- Computationally easy way of generating a "rich" periodic signal with many harmonics.
- Chowning 1973 paper posted on Ctools site.