Intro To:
Audio Synthesis For Music

ALAN WOLFE
HTTP://BLOG.DEMOFOX.ORG

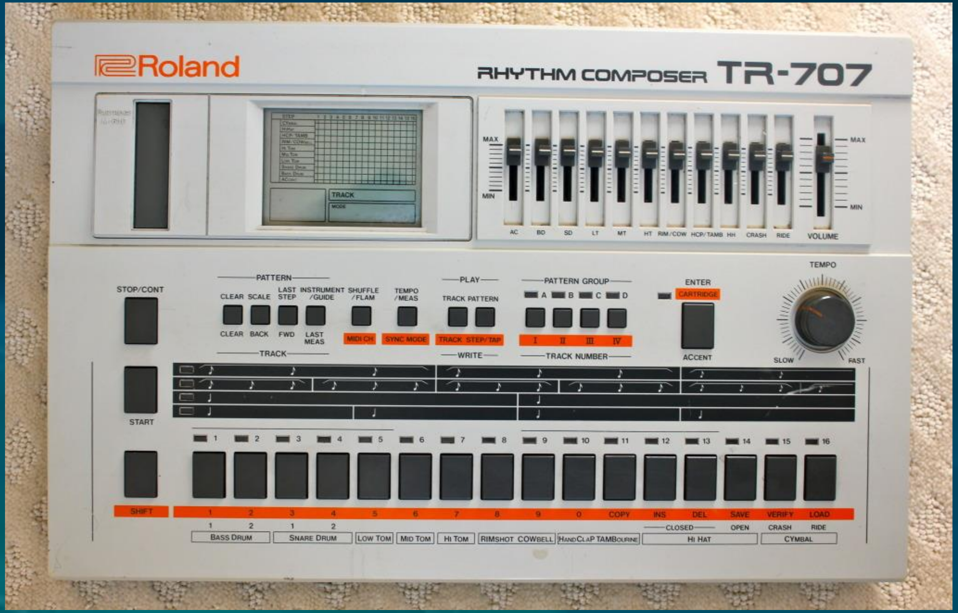
Goals & Intro

- Get an intuitive sense of audio synthesis for music
- Be able to experiment immediately and get results
- Caveat: I am a programmer by trade, not a musician –OR– a sound designer!
- I'm using a qwerty keyboard for demos, but MIDI devices are pretty easy to interface with too!
- Music/Audio/Synth is very much "whatever works"
- More Audio Synth topics on my blog: blog.demofox.org

Roland 303 Synth & 707 Drum Machine

1982

1984



https://www.youtube.com/watch?v=McFMCh8CFag&t=170

Modular Synth

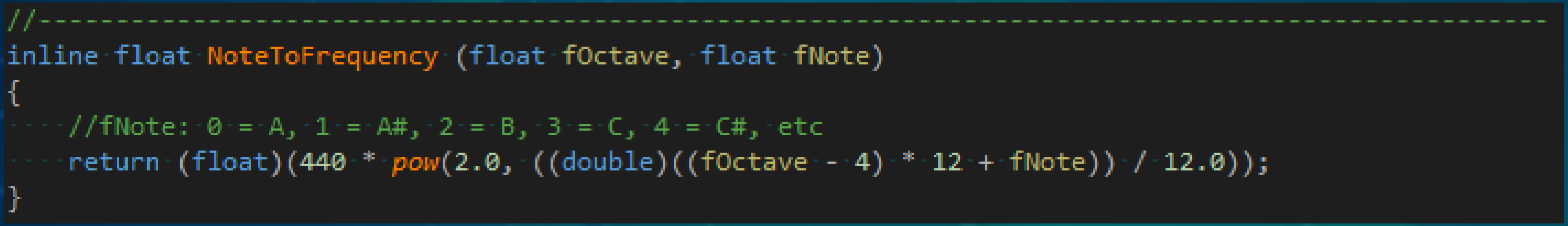
https://www.youtube.com/watch?v=H511Iye6BsI&t=40

What Is Audio Data?

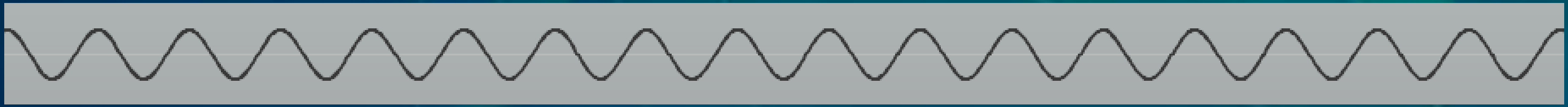
- A stream of numbers between -1.0 and 1.0 (usually)
- Changes in values important, not values themselves
- Value dictates speaker location over time
- Better speakers able to move more quickly and accurately
- .wav file literally = header, then samples
- Resolution:
  - X axis = sample rate (audio samples per second)
  - Y axis = bit depth (size of floating point numbers)
  - Interleaved data for each channel
- **EVERYTHING** is made from sine waves added together (DSP)

What Is A Note?

- A note is just a frequency. Non linear: $freq=440*2^{(note/12)}$
- An octave is a range of 12 notes.
- Double a note frequency to go up an octave.
- Relative notes matter, not absolute.



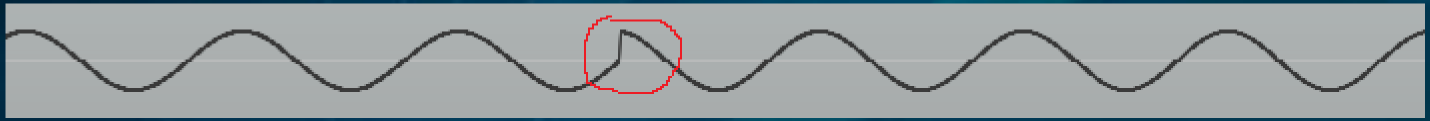
```
//---------------------------------------------------------------------------------------------------
inline float NoteToFrequency (float fOctave, float fNote)
{
    //fNote: 0 = A, 1 = A#, 2 = B, 3 = C, 4 = C#, etc
    return (float)(440 * pow(2.0, ((double)((fOctave - 4) * 12 + fNote)) / 12.0));
}
```



Demo: Sine

Popping

Popping = discontinuity in data. **Usually** unwanted.



Solutions:

1. Free Spinning Oscillator
2. Volume Envelopes (more info later)

Free Spinning Oscillator:

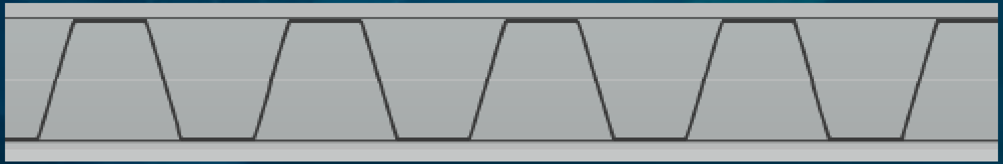
NO: output = sine(time*frequency*2*pi);

YES: output = sine(phase); phase += (frequency*2*pi)/sampleRate;

Demo: Popping

Clipping

Clipping = going outside of [-1,1]. **Usually** unwanted.



- A Form of Distortion
- sine waves clamp, changing sound (frequencies)
- Mostly sounds very bad!
- Sometimes sounds very good! (guitar)

Demo: Clipping

Mixing Multiple Sounds

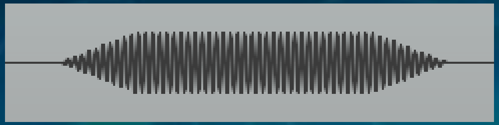
- Just add sample values together!
- Can cause clipping
- Often, peaks and valleys cancel out
- Still a problem to watch out for
- Automated volume adjustments can help (more info later)
- **Terminology:** Polyphonic synthesizer
  - Can play multiple notes at once!

Demo: Mixing

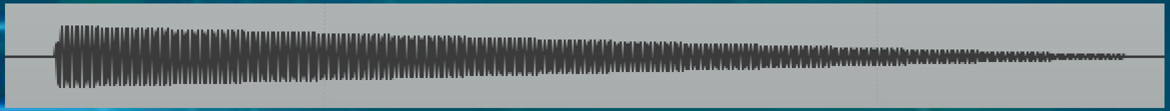
Envelopes

- Volume changes over time
- Can solve popping at beginning and end of sounds
- Can also make things (eg. sine waves) very different!

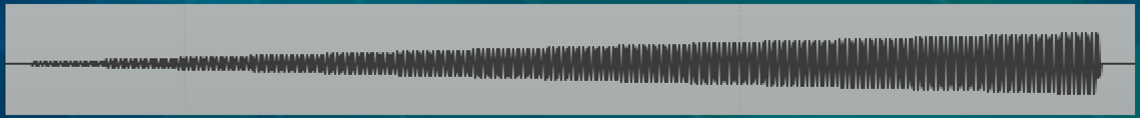
Minimal



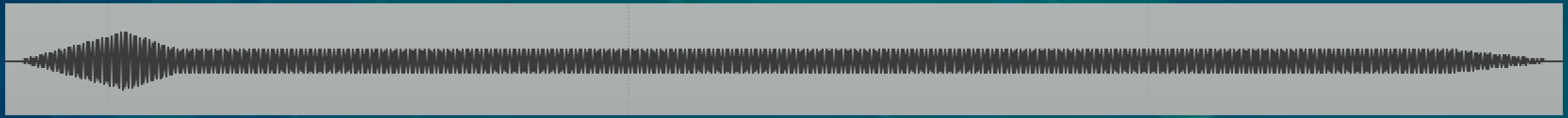
Bell



Reverse Bell



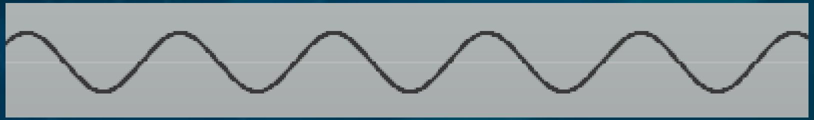
Flute



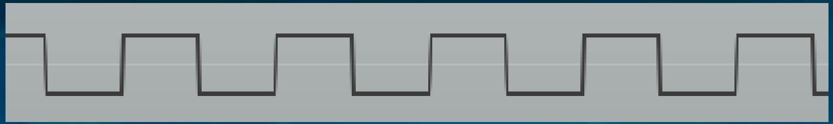
Demo: Envelopes

Common Wave Forms

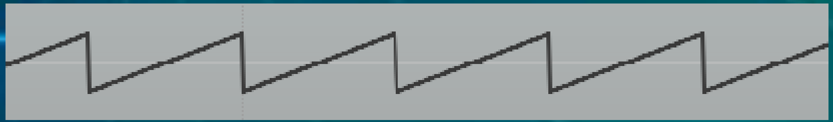
Sine:



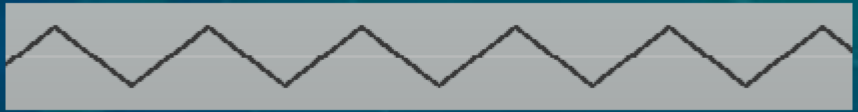
Square:



Saw:



Triangle:



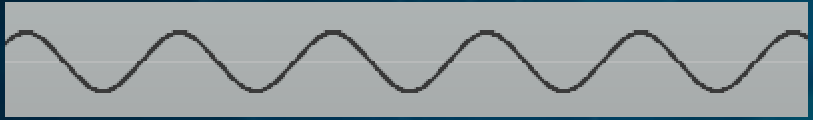
Demo: WaveForms

Aliasing

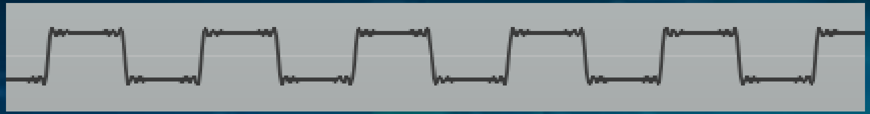
- Another form of distortion (harshness / hurts ears)
- **Terminology:** Nyquist Frequency
  - If you have N samples per second, limited to freq. N/2.
  - Going above that causes aliasing.
- All sound made up of adding sound waves together
- Perfect corners = infinite sine waves at infinite frequencies.
- **Terminology:** Band Limited Signal
  - Limiting signal to a window of frequencies.
- Yes, aliasing is same term and meaning from graphics!

Band Limited Wave Forms

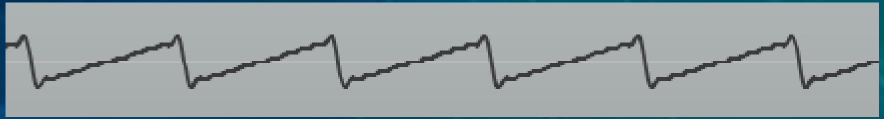
Sine: 1 frequency



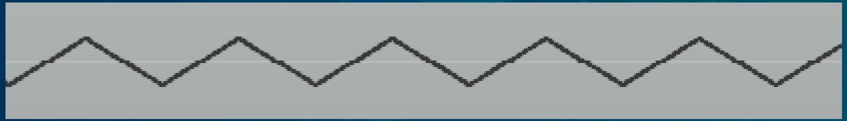
Square: All harmonics



Saw: Odd harmonics



Triangle: Odd harmonics



Terminology: Harmonic
Integer multiples of a frequency

Demo: BLWaveForms

Additive Synthesis

- Add sine waves together to make interesting sounds
- Can analyze sounds in real life and mimic them with limited success (most real sounds are very complex)
- Or, just play around until you have something interesting.

Demo:

- 10 harmonics
- Envelope: 0% quiet, 5% full loud, 10% half loud, 100% quite
- Each harmonic envelope lasts: 1.5s / (index)
- Lower frequencies last longer, like real life sounds.

Demo: Additive

Tremolo & Vibrato Effects

- **Terminology:** Low Frequency Oscillator (LFO)
  - A low frequency wave (usually sine wave), commonly used to drive other effects or behaviors.
  - Very cool and powerful technique!
- Tremolo = use an LFO to control amplitude & volume over time
- Vibrato = use an LFO to control frequency over time

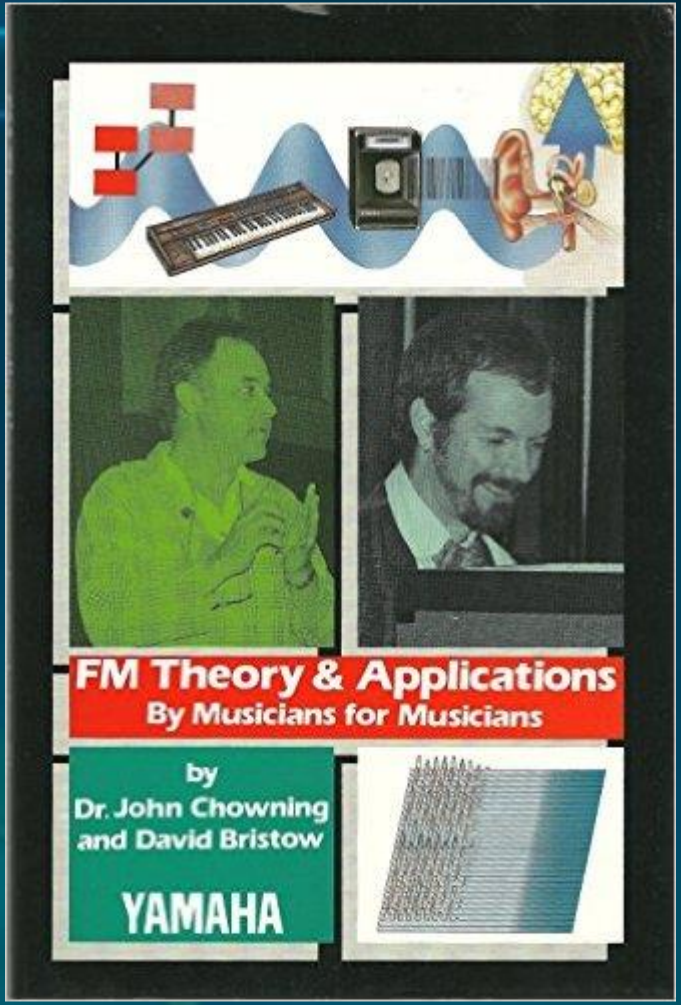
When LFO Taken To Audible Frequencies (HFO?)

- Tremolo = Amplitude Modulation (AM) Synthesis
- Vibrato = Frequency Modulation (FM) synthesis

Demo: TremVib

FM Synthesis

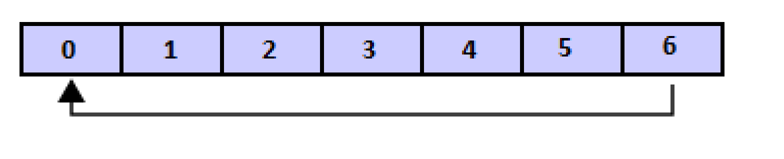
- Bring Vibrato frequency up to audible frequencies
- Very hit or miss black magic, but based on solid math.
- 2 Parameters:
  1. Frequency of frequency altering wave (frequency)
  2. How much the frequency should be altered by (amplitude)
- Can use trees of FM to create complex sounds.
- Very Deep Topic!
- Yes, related to FM radio a bit.



Demo: FMSynth

Delay (Echo) Effect

- Circular buffer of audio samples.
- Parameters:
  - Time: determines size of buffer.
    - bufferSize = delayTime * sampleRate * numChannels
  - Feedback: before writing (adding) a new sample into the circular buffer, multiply the existing sample by this value.
- How it Works:
  1. ret = buffer[index] + inputSample;
  2. Buffer[index] = buffer[index] * feedback + inputSample;
  3. Index = (index+1) % bufferSize;
  4. return ret;



Demo: Delay

Multitap Reverberation Effect

- Reverb = many echos at different volumes and times, from a real environment. Hallways, caverns, churches, small rooms, etc.
- Correct Reverb = Convolution of impulse (single 1.0 sample, or a clap) in said environment.
- Simpler / Cheaper to just do N echoes at various times and amplitudes.
- Multitap Reverb = a delay effect with lots of reads into the delay buffer, multiplying each read by a volume scaling value. Approximates convolution! Not as high quality though.
- Demo: 7 taps in a 0.662s delay buffer at various volumes.

Demo: Reverb

Flange Effect

- Discovered in analog days by putting a finger on a tape to slow it down.
- Mix a sound with itself, at a different time. Time delay varies over time.
- Implementation: Use a delay buffer, but instead of reading from the write head, offset the read from the write head using an LFO.
- Parameters:
  - Time: determines size of delay buffer
  - Frequency: how fast the read offset should move between 0 and bufferSize.

Demo: Flange

Synth Drum + Percussion Sounds

- You are better off using sound samples!
- There are some ways to make it sound passable though.
- This demo is not so great, and just scratches the surface.
- Drum:
  - Use sine wave. Envelope = 10ms attack, 10ms hold, 175ms decay.
  - Make frequency drop over time.
- Cymbals:
  - Shape noise (static) with an envelope.

Demo: Drum

Volume Adjustment Effects (Simplified)

- Compressor = Slowly turn down volume if too loud, turn it back up again when it’s quieter.
- Limiter = If sound clips, turn down sound instantly to make it not clip. Turn sound up again later when quieter.
- Ducking = make something quieter to make something else stand out more.
- Several games do the above to fight ear fatigue and to make sounds stand out.
- See also battlefield for high dynamic range audio.
- In music, it’s common to use “side chain compression” to make a compressor turn down music in response to drum sounds being loud.

Demo: Ducking

Filtering Effects & Subtractive Synthesis

- Some filters can change volume of specific frequencies
- Low pass filter (LPF) and High pass filter (HPF) as examples
- Subtractive Synth = carve away frequencies
  - Can only cut away what's there (odd / all harmonics matters now!)
- Subtractive synth can make interesting sounds, as well as bandlimited wave forms.
- You can also put filter parameters on LFOs for more interesting sound scapes.
- More info: read about Biquads

Demo: Filtering

Stereo Effects

- Everything has been “mono” til now. 1 channel.
- You can do interesting things with multiple channels.
- Positional sounds, varying effects per ear, and more.
- Positional Sound:
  - Deep topic, but can be faked by putting left ear volume on sine LFO and right ear volume on cosine LFO.
- Ping Pong Delay:
  - Echoes first to one ear, then the other, then feeds back into first ear.

Demo: Stereo

Musical Audio Synth Cheat Sheet

Synthesis Types:

- Additive – add sine waves together
- Subtractive – filter away frequencies
- FM – modulate frequency by another wave
- AM – modulate amplitude by another wave

Effects:

- Envelopes / Ducking – Adjust volume over time
- Tremolo – Adjust volume by LFO
- Vibrato – Adjust frequency by LFO
- Delay (echo) – Ring buffer
- Multitap Reverb – Multiple reads from ring buffer
- Flange – Delay ring buffer read offset by LFO
- Filtering – Change frequency amplitudes. LFO?
- Limiter / Compressor – automatically adjust volume.

Wave Forms:

- Sine
- Saw
- Square
- Triangle
- Bandlimited Forms
- Noise

Avoid:

- Popping – Keep it continuous
- Clipping – Stay in [-1,1]
- Aliasing – No frequencies too high

Get The C++ Demo Program and start hacking! https://github.com/Atrix256/MusicSynth
More Audio Synth on my blog: http://Blog.demofox.org