Music and Engineering: Musical Instrument Synthesis

Tim Hoerning

Fall 2008

(last edited on 11/19/08)

Outline

- Early Electronic & Electro-mechanical Instruments
  - Hammond Organ, Mellotron, Theremin, etc
- Fundamentals (Building Blocks)
- Synthesis techniques
  - Additive Synthesis
  - Subtractive Synthesis
  - Distortion Synthesis
  - Synthesis from analysis
  - Granular Synthesis
  - Physical Modeling
- Representations for Musicians

Electromechanical Instrument

- Several Famous instrument were created with using coils similar to electric guitar pickups and a tone generators
  - The Fender Rhodes electric piano used a piano like action to strike metal tines (small bars) to generate a pitch
  - The Hohner Clavinet used a tangent connected directly to a key to strike a string which was generated a pitch for a pickup.
    - Musical Example: Superstition by Stevie Wonder
  - The Hammond B3 used a rotating varying reluctance tone wheel positioned above a pickup to generate the smooth organ sounds.
  - The Mellotron actually used loops of tapes to produce the notes
    - Musical Example: Sgt. Peppers album by The Beatles

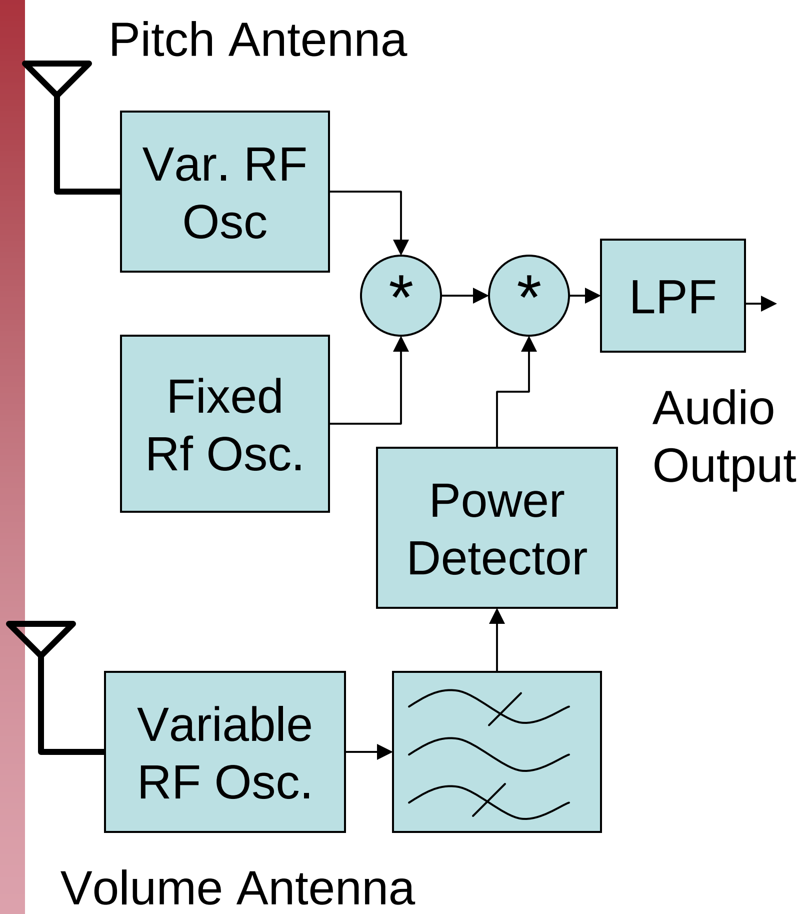
Fully Electric Instruments

- Some older Organs used large banks of vacuum tube oscillators connected to a conventional organ keyboard
  - Hammond NovaChord
  - Allen Organ
- One of the first completely electronic instruments was the Theremin
  - Invented less than 20 years after the invention of vacuum tubes
  - Unique interface required musicians to play without touching the instrument
    - Two antennas were used
      - The upright antenna controlled the pitch. The closer to the antenna, the higher the pitch
      - The horizontal loop antenna controlled the output volume. The closer to the antenna the quieter. This allowed notes to be plucked.
    - Very difficult to play
      - The extreme sensitivity required the user to hold their body steady while playing so as not to affect the pitch
      - Clara Rockmore was the only person to tour exclusively as a Theremin player
    - Mostly used for sound effects
- Other instruments were created around non-standard interfaces
  - Ribbon controller
  - Electro-Theremin, (Tannerin) – sounds like a Theremin, but easy to control.

Musical Examples:
Edison's Medicine - Tesla

Musical Examples:
Good Vibrations – Beach Boys

The Theremin



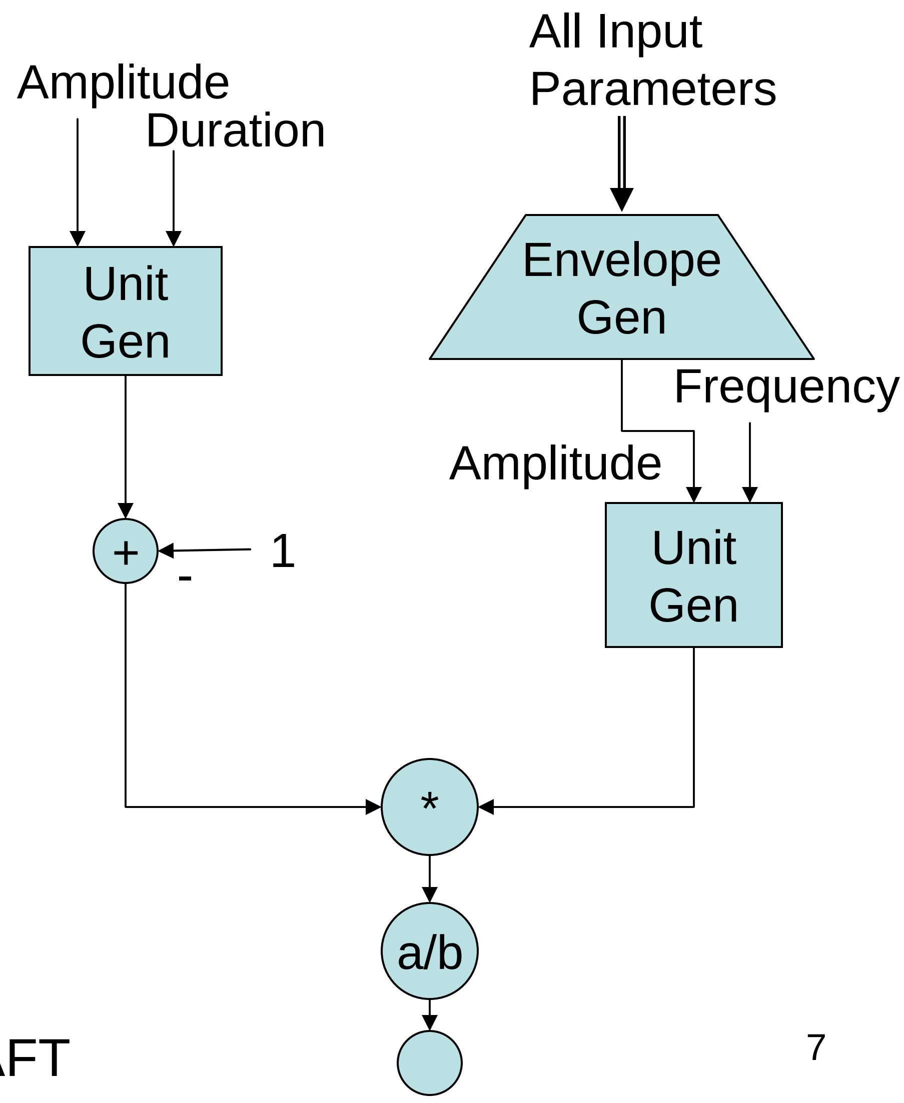
- The Theremin utilizes two RF devices (typically ~ 300kHz)
  - One has a fixed frequency
  - The other has a variable frequency determined by the antenna
- These are beat against each other (heterodyned) to generate an audio output.
- Another variable oscillator can be used to create a volume control (not always present on simpler modern Theremins)

Computer Synthesis Building Blocks

- Instruments are implemented as algorithms typically using a specialty software package
  - Could be in a rack mount synthesizer
  - Or a general purpose computer
- Synthetic Instruments are often built up from *Unit generators*.
  - Simplifies the technical details for musicians
  - UGs are interconnected to form instruments
  - UGs are often modeled graphically so than an instrument flowchart

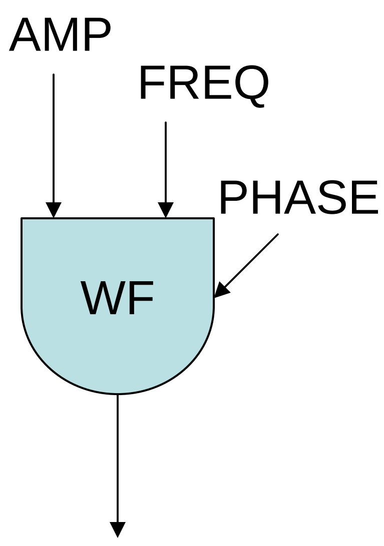
Signal Flowchart

- Behaves like a simplified “digital circuit”
  - Output can be tied to more than one input
  - Outputs can never be tied together
  - Can combine outputs through mathematical operations
    - Addition (+) is used for mixing audio signals
    - Subtraction (+ with the negative input labeled with a – sign) Combining two signals while inverting one.
    - Multiplication (*) is typically used for amplification of a constructed signal
    - Division (a/b) is typically used for attenuation of a constructed signal
- Output is defined a small empty circle



Oscillator

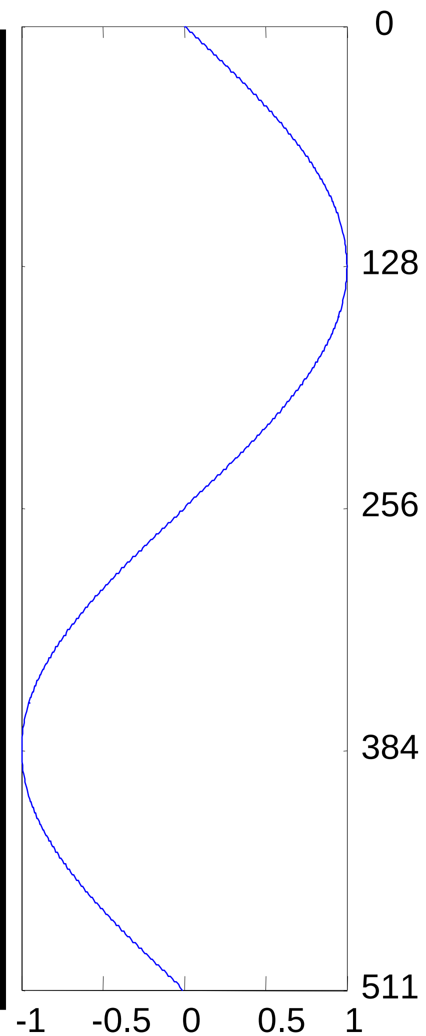
- Most fundamental *UG* is the oscillator
- Symbol inside generator describes type (sine, square, general waveform)
- Inputs generally given short representative input names
  - AMP = peak amplitude
  - FREQ = frequency
    - Number of Hertz
    - Sampling Increment (SI)
  - PHASE = starting point in the cycle



Implementation

- Direct Evaluation
  - Compute while generating
  - Very slow for most synthesizers
- Wavetable
  - Stored waveform (buffer in ROM)
  - Contains one period of the waveform
    - Later Romplers may contain more complete samples of actual musical instruments
  - Starting Sample is determined by the Phase input

0	0
1	0.0123
	...
127	0.9999
128	1.0
129	0.9999
	...
255	0.0123
256	0.0
257	-0.0123
	...
383	-0.9999
384	-1.0
385	-0.9999
	...
510	-0.0245
511	-0.0123



Sampling Increment

- To generate the “fundamental” of the wave shape, read out at the sampling rate
- Harmonics can be generating ready every other sample (octave) or other multiples (i.e. every 3rd sample = fifth above octave)
- Other frequencies can be created by specifying a Sampling Increment (SI)

$$SI = N \frac{f_o}{f_s}$$

Fractional Indexes

- The SI is likely not to be an integer
- Three methods exist for using fractional SI's while reading out the waveform from the wavetable. The complexity increases in this list
  - Truncation – round down to the nearest integer
  - Rounding – round to the nearest integer (up or down
  - Interpolation – Estimate the value at this time via a linear interpolation (or more complex interpolation)

SNR Effects of 3 methods

$$k = \log_2 N$$

- Consider the following table where N is the number of elements in the table
- SNR is approximated by the following equations
  - Truncation = 6*k* – 11dB
  - Rounding = 6*k* – 5 dB
  - Interpolation = 12(*k*-1)dB
- For the 512 element example table this yields 43, 49 & 96 dB respectively
- This SNR would need to be combined with D/A SNR to get a true estimate of the effect on the quality
- This illustrates a implementation between computational power and memory usage.

Other Methods of Defining Waveforms

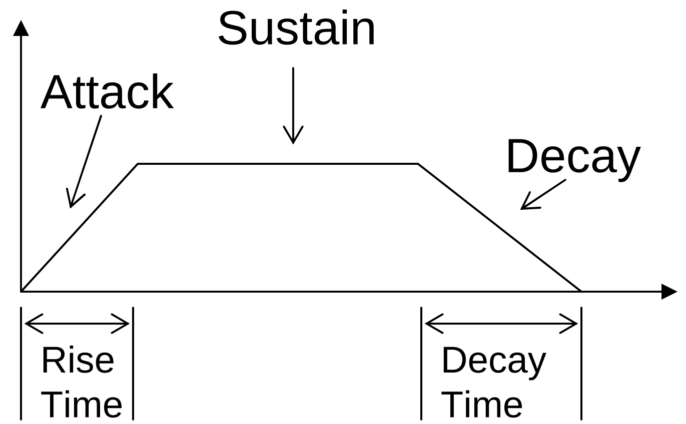
- Besides direct evaluation or stored wavetable, The waveform can be described with a piece wise linear evaluation
  - This is defined as a set of *breakpoints*
    - Points in time and amplitude that dictate where the waveform changes slops
    - A line is drawn between the breakpoints to determine the waveform
    - All points are described as a phase and the amplitude at that phase.
    - After generation these functions are usually stored in a RAM wavetable.
  - Problems can arise from a harmonically complex waveform being generated with a high frequency fundamental
    - The upper harmonics may exceed the Nyquist rate (fs/2) and create images in the frequency domain.
    - This would generate an in harmonious instrument

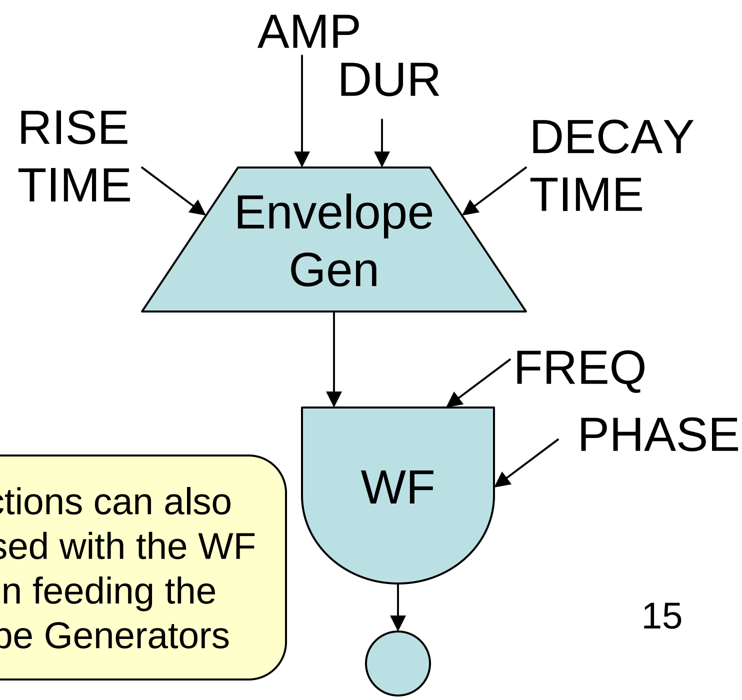
Define in the Frequency domain

- To combat the possible introduction of upper harmonics that will create images, one can specify the waveform in the frequency domain
  - Waveforms are defined a series of data structures where each structure element includes the
    - Amplitude
    - Partial Number
    - Phase
  - Partials above the Nyquist rate can be eliminated by not adding that partial to the rest during the synthesis phase of the process.

Functions of Time

- It is often desired to make an oscillator vary it’s amplitude with time
- This will modify the “envelope” of the signal, hence their name of envelope generators
- The Envelope generator is connected to the AMP input of the *UG* to modify the amplitude

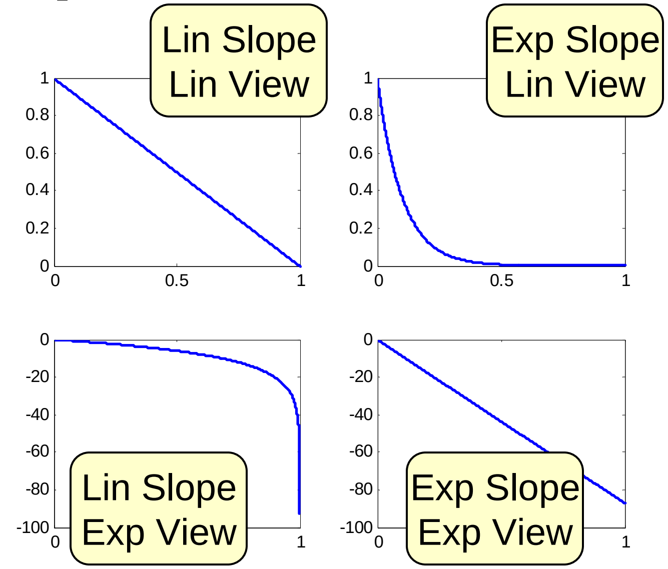


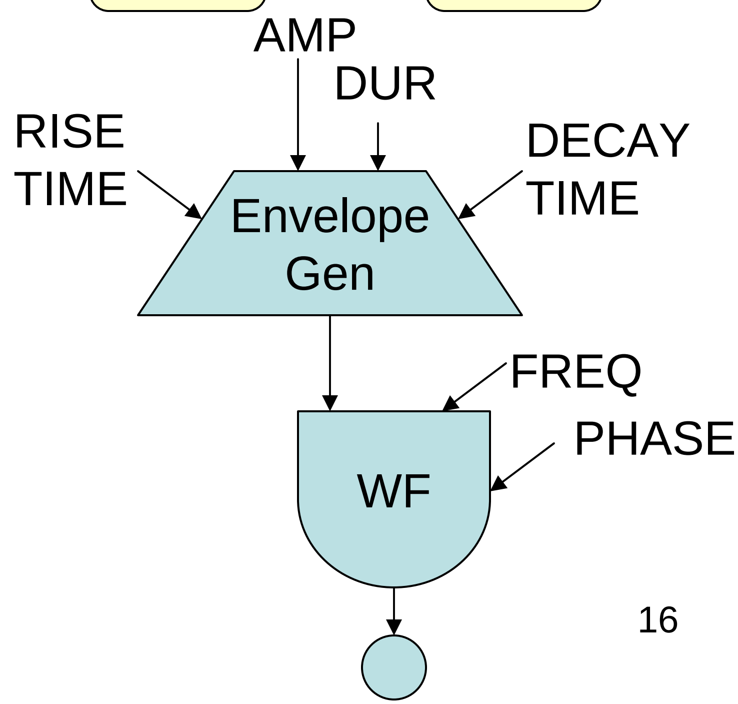


Connections can also be reversed with the WF function feeding the Envelope Generators

More Envelopes

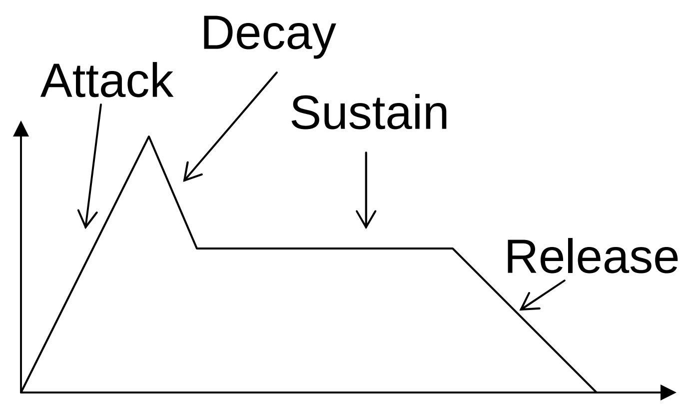
- The function describing the segments of the envelope can be linear or exponential
- Both are useful for different modeling purposes
  - Exponential is the method by which natural instruments die away.
  - Linear is useful for the sustain region and slow attack times
- The envelope can have a great effect on the timbre of the sound
  - Short attacks are more common in percussion
  - Long attacks are more commonly found in acoustic instruments such as a pipe organ.





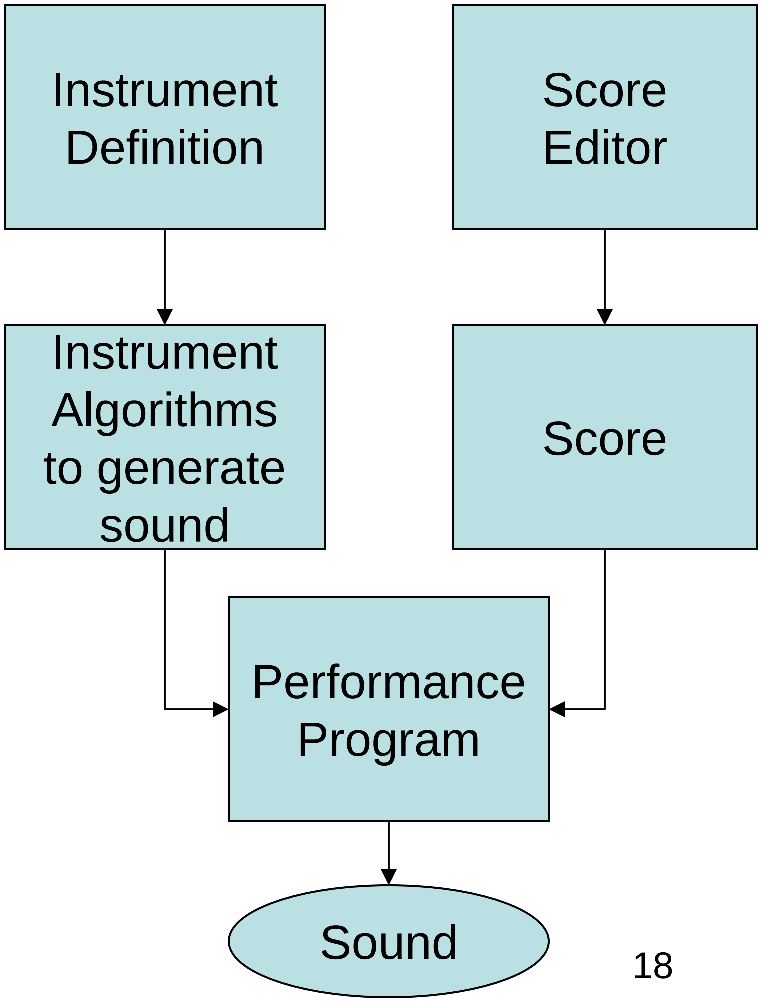
Additional complexity

- Can add another segment to the envelope to better match more instruments
  - Section added after attack to simulate the fast die out of a struck note before the sustain portion – This section steals the name Decay
  - Decay section at the end of the wavefrom is renamed “Release”

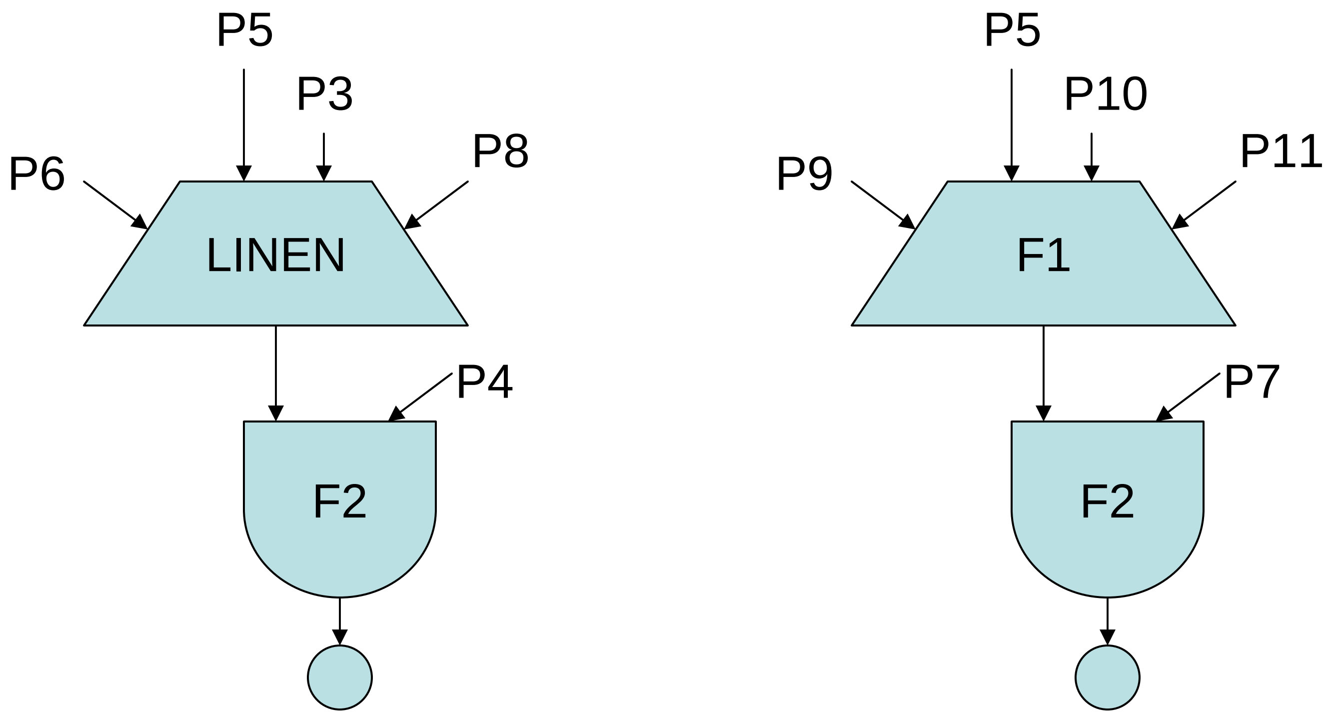


Programming languages

- Before GUIs and HW synthesizers, there were software languages for generating computer music
- Csound and Cmusic are the two descendants of the first packages designed to create sound on workstations
- Like any good programming environment, the tasks are build up in stages. The sound definition is used in parallel with the music definitions. This keeps the code cleaner



Csound vs. Cmusic




```
instr    1
k1       linen    p5,p6,p3,p8
a2       oscil    k1,p4,2
out      a2
endin
```

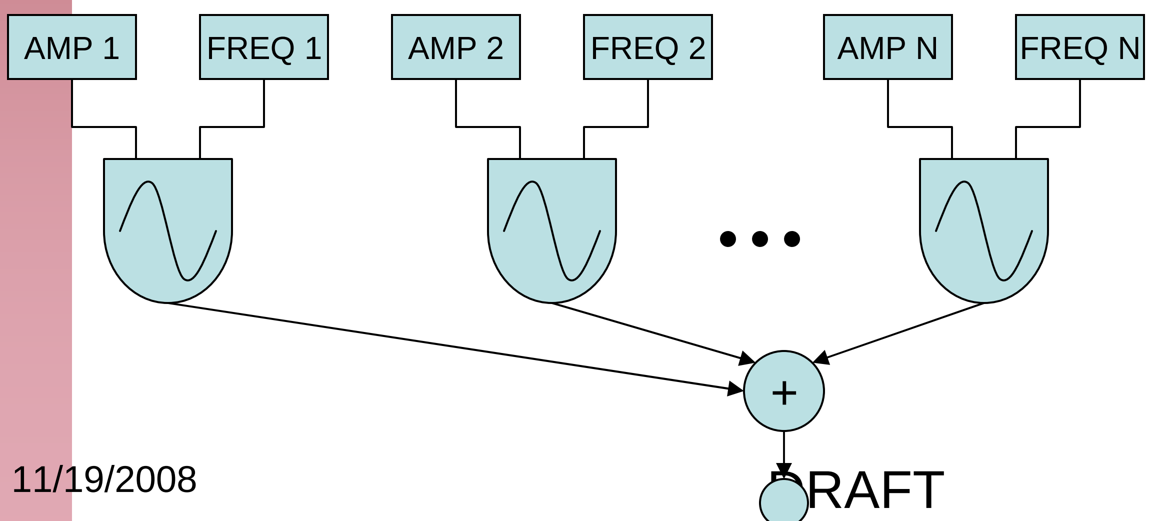
k1 = env gen output, p5=amplitude of note, p6 = rise time, p3=duration,p8=decay time,p4=frequency,2=type of waveform

```
ins      0      SIMPLE;
osc b2 p5 p10 f3 d;
osc b1 b2 p6 f1 d;
out b1;
```

b2 = 1st oscillation output, p5=amplitude of note, p10 = dur, f3=function to control envelope shape, d=phase of oscillator, p6=frequency, f1=waveform pattern to generate

Additive Synthesis

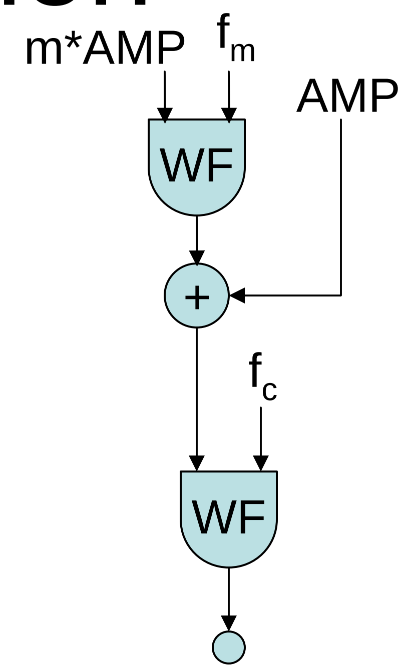
- Previous diagrams were fine for describing steady state tones, but couldn't match transients
  - Harmonics all arrived and departed at the same time
  - Higher frequencies were perfect – no adjustment for out of tune
- New Model (shown below) represents every component with its own set of sine wave UGs
  - Adding all the outputs gives the desired sound – Additive Synthesis
  - Often called *Fourier recomposition* – uses synthesis by analysis
  - Can combine multiple instruments, but care should be taken to align temporal peaks
  - Requires significant computational resources to generate one sound
  - Required multiple configurations to support different intensity levels (instruments sound different depending on the force of the physical attack



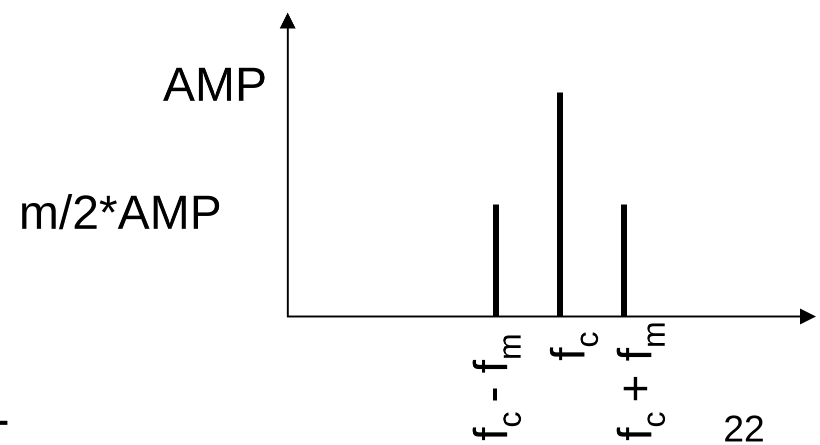
Modulation

- Modulation is alteration of the following
  - Amplitude
    - Amplitude modulation
      - Basically tremolo. A signal source is connect to the Amplitude input of the audio generator
    - Ring modulation
      - Moves result to a different frequency center (same process as in the ring modulator effect from the last lecture)
    - Single-sideband modulation
      - Not discussed – a radio method with little use in music
  - Frequency
    - vibrato

Amplitude Modulation



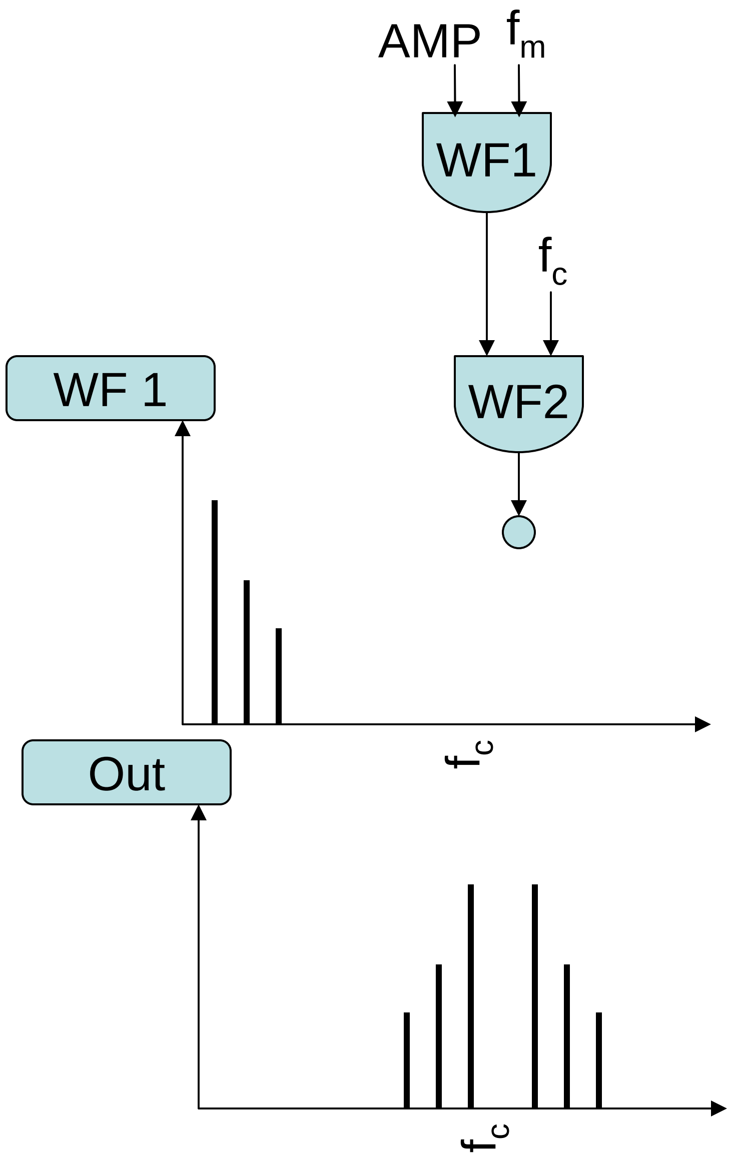
- Basically tremolo. A signal source is connect to the Amplitude input of the audio generator
- Generates side bands
- Perception
  - < 10Hz – ear tracks amplitude variations
  - 10Hz < x < critical band boundaries – user hears amplitude of the average of the output
  - > 1/2 critical band – perceived as additional tones



Ring Modulation

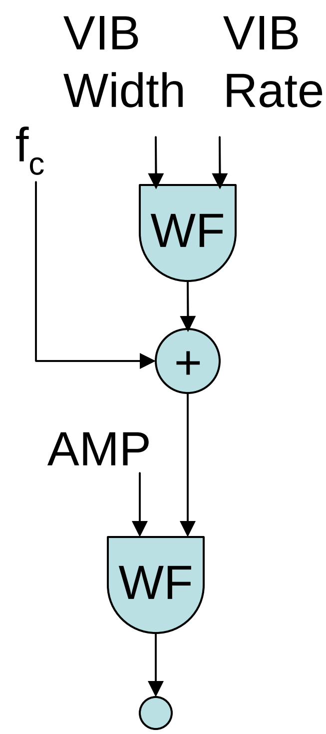
- Multiplies two waveforms together to spectrally dense signal also called
  - Balanced Modulation
  - Double Sideband Modulation
  - Called mixing in the RF field
- Produces outputs at $f_c + f_m$ and $f_c - f_m$
- Can use multiply to generate RO instead of 2 oscillators
- If either oscillators are zero – no output
- If both waveforms have *p* and *q* harmonics respectively, the output contains 2**p***q* harmonics (all possible products of the harmonics

$$RM(x) = \cos(Ax) \cdot \cos(Bx)$$



Frequency Modulation

- Applies a small shift to the frequency center
  - Average is still center frequency, but pitch varies around it
  - Modulation usually at most a few percent of the center frequency
  - Modulation rate is below the audio range
  - Higher rates lead to frequency modulation synthesis



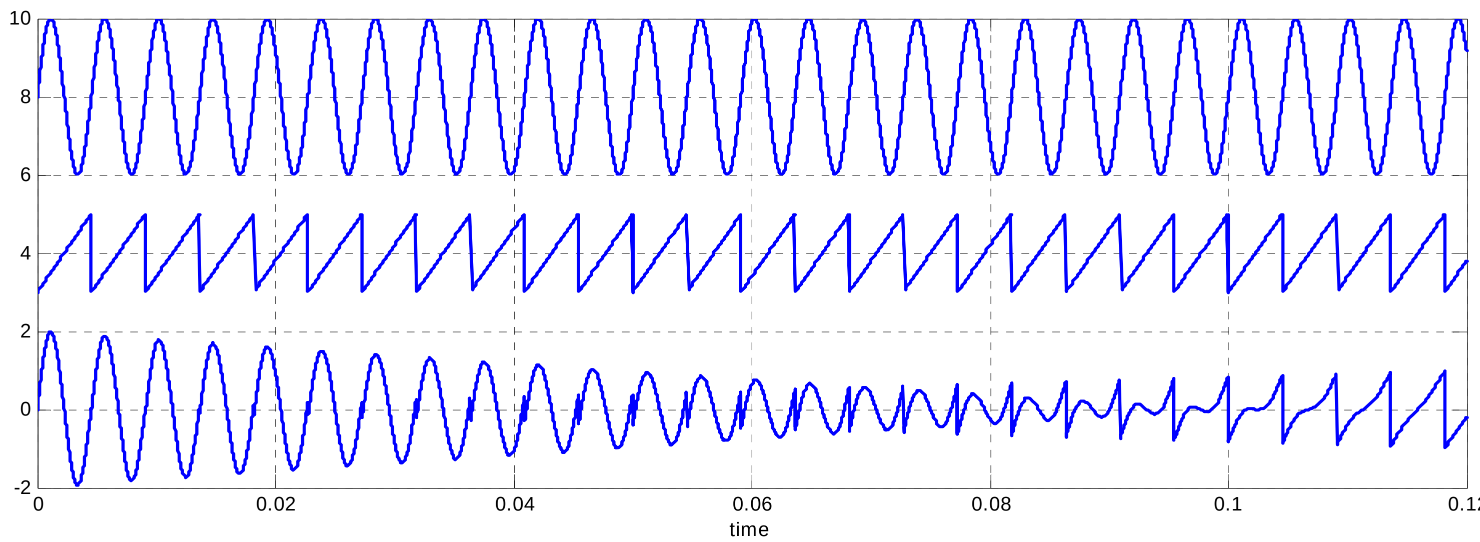
$$FM(x) = \cos(A + \cos(Bx))$$

Noise Generators

- Generate a Distributed Spectrum
  - Fills many bands
  - White noise is flat across all bands
  - Generated by a random (or pseudo random) number generator
  - When random samples are picked at a rate < the sampling frequency, the high end is rolled off

Spectral Interpolation

- Implemented by using a mixer to gradually switch between two sounds
  - With mix value set to 0 all of sound 1
  - With mix value set to 1 all of sound 2
  - With mix value set to 0.5 – 50% of sound 1 and 50% of sound2



Distortion Synthesis

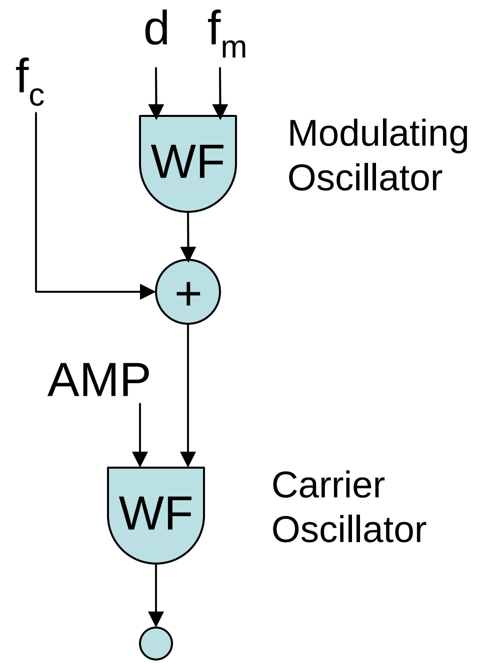
- Additive Synthesis required too much computational complexity
- Non Linear methods were introduced to allow a wide range of sounds while keeping complexity down
  - The spectral complexity increases with distortion.
- Several Methods are commonly used
  - Frequency Modulation
  - Nonlinear Wave-shaping
  - Discrete Summation Formulas (not covered)

FM Synthesis

- Early FM synthesis research was lead by J. Chowning in the mid to late 1970s
- FM synthesis saw widespread use in PC sound cards before the falling price of memory made wave table based cards more affordable
- Unlike the vibrato example on a previous slide, now the modulation is in the audible range.
  - The can yield non-harmonic results caused by the modulation process.

FM Synthesis

- Typically only used Sinusoids for oscillators since more complex signals produce more complex spectra
- d=deviation = max (f_m) – min (f_m)
  - Instantaneous frequencies are f_c-d to f_c+d
  - When d=0, the output is sinusoidal
  - If d>f, negative frequencies result
    - Requires processor to output sample in reverse to show phase change
    - Frequency is folded over to positive axis with a phase change.

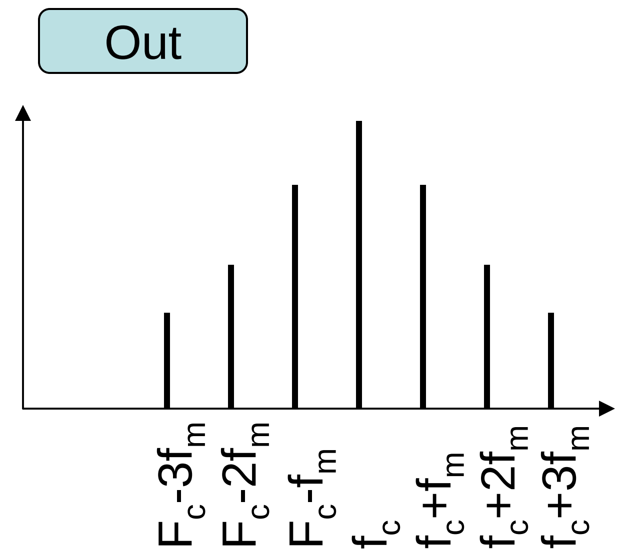


FM Synthesis Spectra

- Using Sinusoids, the output spectrum will look similar to the one at left
- Frequencies present are $f_c \pm kf_m$ where *k* is a natural number.
  - Power division depends on *d*
    - *d*=0 means all power is in f_c
    - As d increases, k increases and more power is added to the sidebands
  - Define the Index of Modulation

$$I = \frac{d}{f_m}$$

$$FM(x) = \cos(d \cdot \cos(f_c x))$$



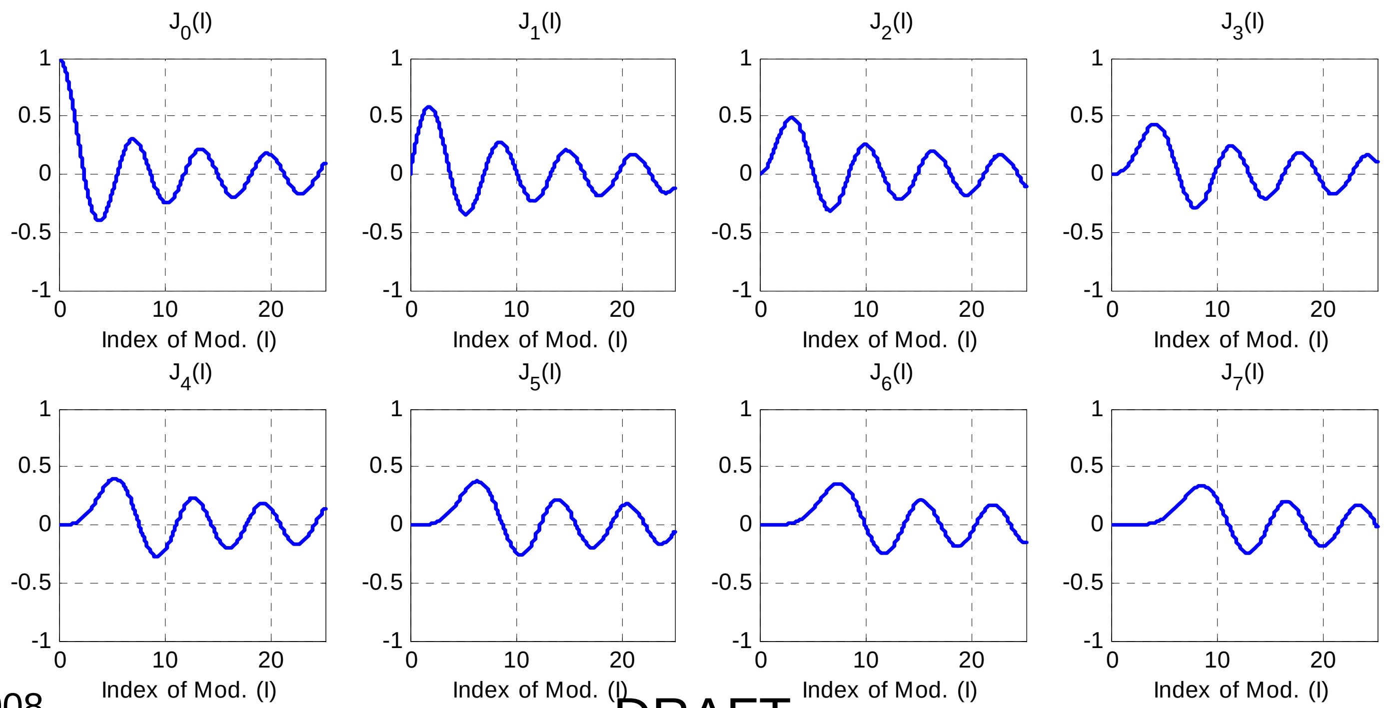
Bessel Functions

- The index of modulation determines the amplitude of each of the side bands according to the Bessel functions listed in the chart
- The sign (phase) of each component is not audibly significant unless there is spectral folding and a wrapped negative component cancels a positive component.
  - Then the two components must be added.
  - Remember that folding negative components to the positive frequency also flips their sign

k	Freq	Amp	Freq	Amp
0	f_c	$J_0(I)$		
1	f_c-f_m	$-J_1(I)$	f_c+f_m	$J_1(I)$
2	f_c-2f_m	$J_2(I)$	f_c+2f_m	$J_2(I)$
3	f_c-3f_m	$-J_3(I)$	f_c+3f_m	$J_3(I)$
4	f_c-4f_m	$J_4(I)$	f_c+4f_m	$J_4(I)$
5	f_c-5f_m	$-J_5(I)$	f_c+5f_m	$J_5(I)$
etc	f_c-kf_m	$(-1)^kJ_k(I)$	F_c+kf_m	$J_k(I)$

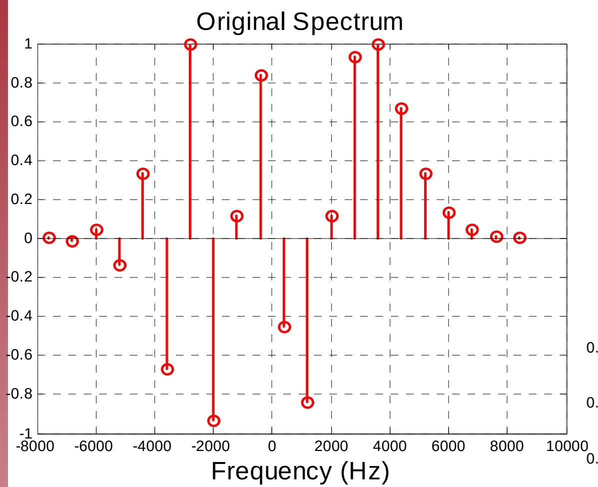
Bessel Functions

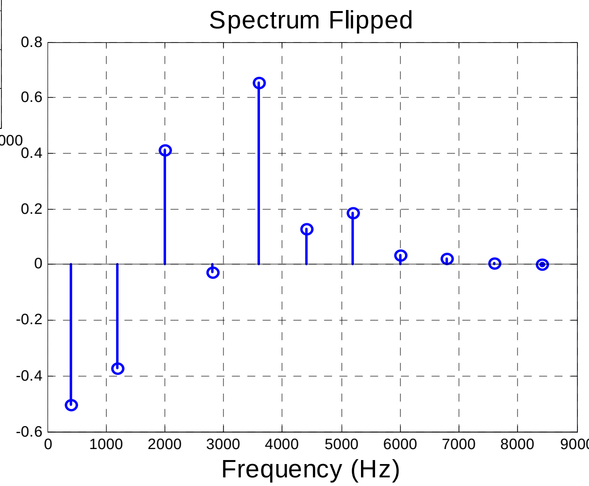
- Plots of the first 8 Bessel functions are shown below.
- Note that for I=0, the only frequency present is the carrier.
- A Rule of Thumb: Only sidebands up to k=I+1 contain significant power (from Jerse)

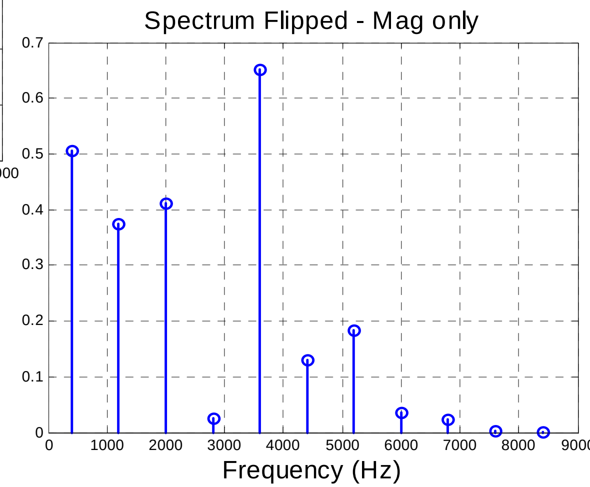


Folded Spectrum Example

- f_c=400Hz, f_m= 400Hz, I=3;

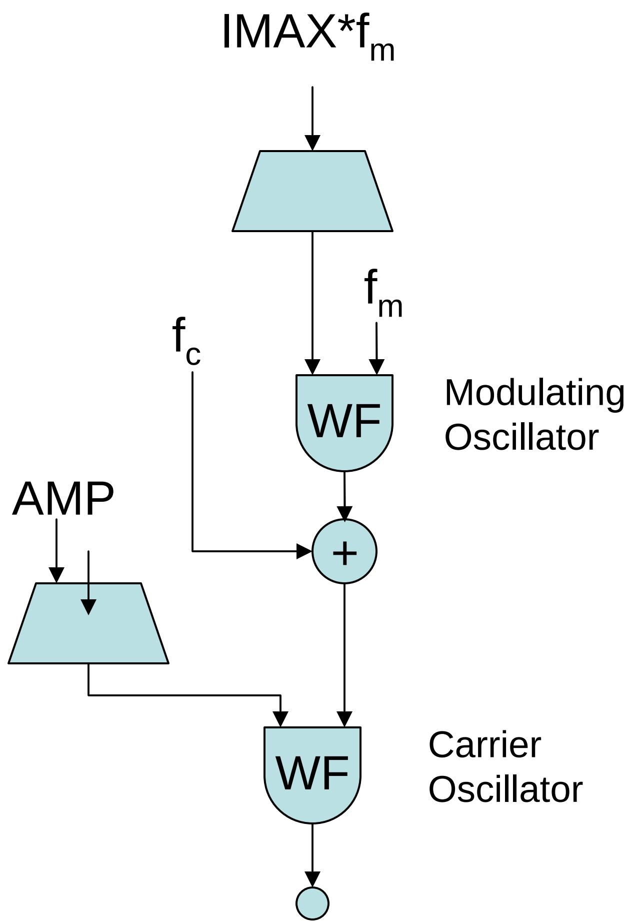






Dynamic Spectra

- In order to have the spectrum evolve as a function of time, provide a envelope control to the d parameter.
- Two different envelope generators are used
  - One for the overall envelope of the sound
  - One for the evolution of the spectrum
    - IMAX is the maximum deviation
    - Does not allow a specification of a specific spectral evolution, but a varying amount of richness



Example Instruments

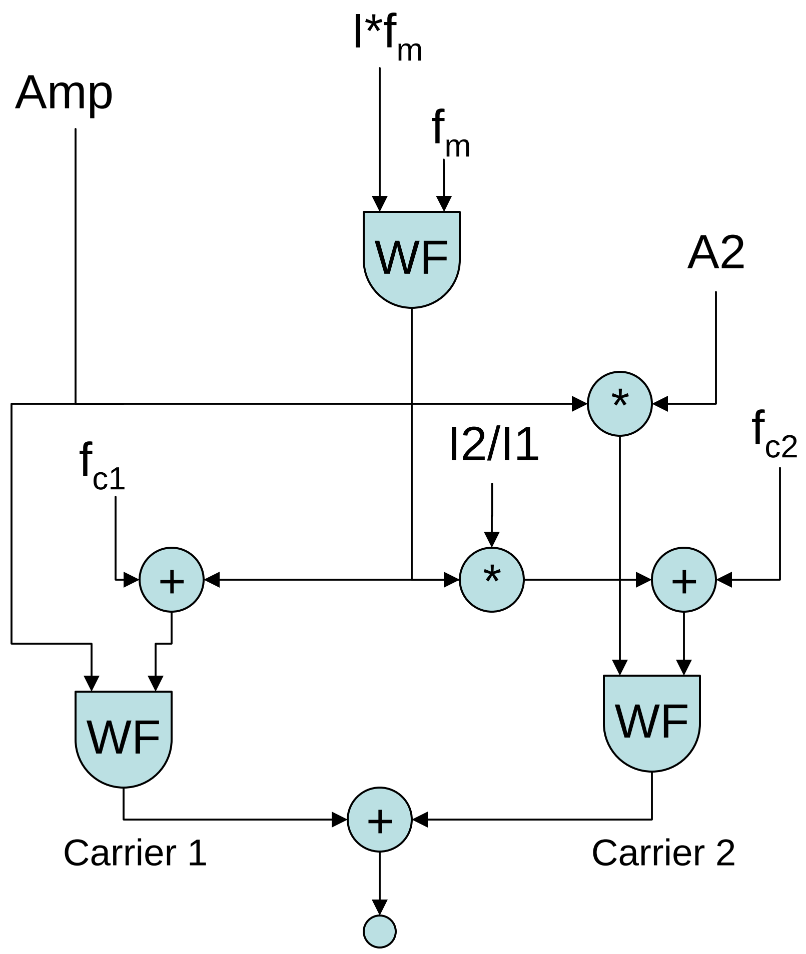
- See Section 5.1D of Jerse.
  - Bell
  - Wood Drum
  - Brass
  - Clarinet

Double Carrier

- Useful in mimicking the formant (fixed resonant frequency) present in acoustic instruments that isn't captured with Single Carrier FM synthesis.
- Two carriers are at fundamental and first formant frequency.
- IMAX is maximum modulation
  - I2 is the ratio of the 2nd carrier to the first. Usually pretty small
  - A2 is usually less than unity too
- Fc2 is usually chosen as the harmonic of the fundamental closest to the formant.

$$f_{c2} = nf_0 = \mathrm{int}\left(\frac{f_f}{f_0} + 0.5\right) f_0$$

- Used by Morrill in synthesis of trumpet tones.



Double Carrier Example Instruments

- See Section 5.1F of Jerse.
  - Trumpet w/ Vibrato
  - Soprano Voice

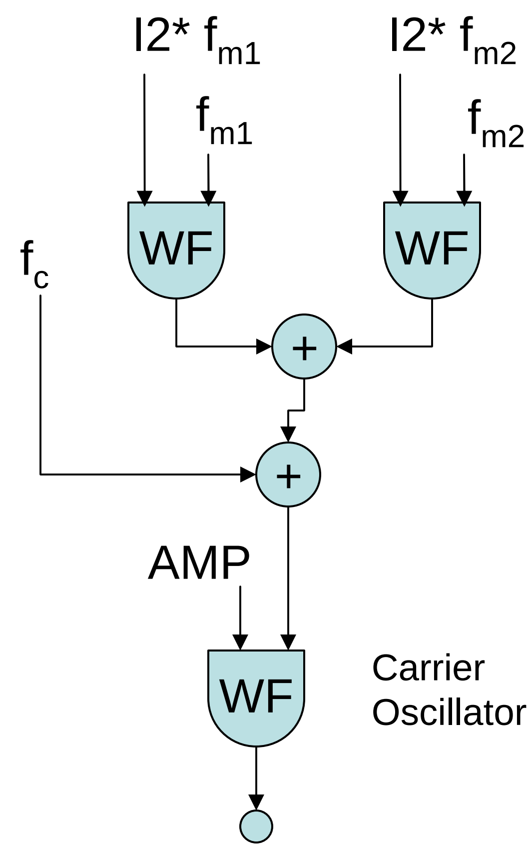
Complex Waveforms

- Example shows sine modulated by waveform with 2 spectral components
  - Frequencies in the output are

$$f_c \pm i f_{m1} \pm k f_{m2}$$

  - Amplitude of the resulting sidebands are determined as the product of Bessel functions

$$A_{i,k} = J_i(I_1) J_k(I_2)$$

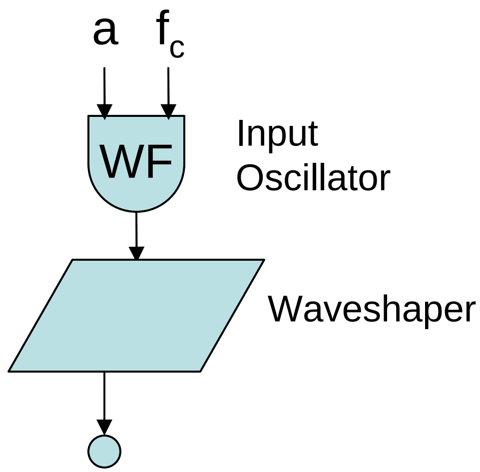


Complex Modulation Example Instruments

- See Section 5.1H of Jerse.
  - Violin

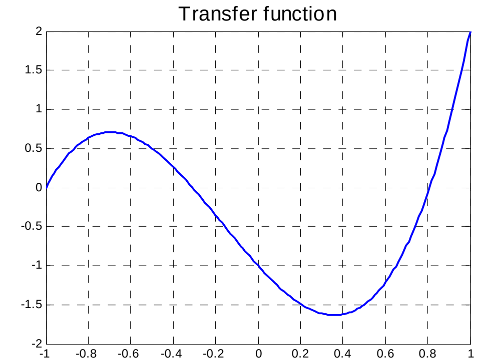
Synthesis by Waveshaping

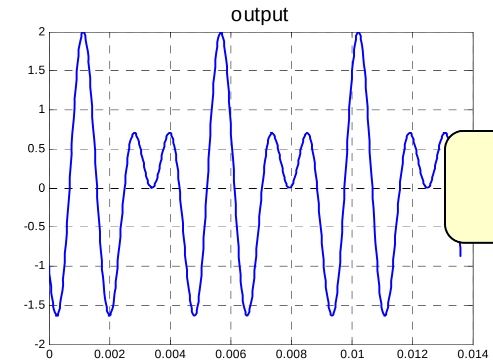
- A different type of non-linear processing
  - Similar to FM
    - is more efficient than additive
    - Dynamic evolution in spectral complexity
  - Unlike FM
    - Can generate a band-limited spectrum



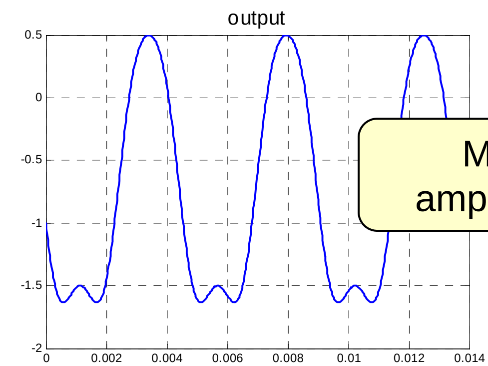
Waveshaping through non-linear transfer functions

- Waveshaping uses the same concept of a transfer function that we saw when considering distortion effects.
  - The output shape will depend on the input amplitude
  - The shape of the transfer function will determine the richness of the output
    - Discontinuities add high frequency components.
    - Standard Symmetry rules apply
      - Odd functions only contain odd harmonics
      - Even functions only contain even harmonics





Max input amplitude =1



Max input amplitude = 0.5

Polynomials

- In order to keep the waveshaping problem tractable, limit the transfer functions to polynomials

$$F(x) = d_0 + d_1 x + d_2 x^2 + \cdots + d_N x^N$$

- This guarantees that the output spectrum will not have frequencies greater that $N*f_0$
- For any given single term polynomial, x^N the ratio of power in the harmonics is given in the table on the following slide.

Harmonic levels

	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}
x^0	1											
x^1		1										
x^2	1		1/2									
x^3		3/4		1/4								
x^4	6/8		4/8		1/8							
x^5		10/16		5/16		1/16						
x^6	20/32		15/32		6/32		1/32					
x^7												
x^8												
x^9												
x^{10}												
x^{11}												

Example:

$F(x)=x^5$

$h_1 = 0.625$

$h_3 = 0.3125$

$h_5 = 0.0625$

Check on adding to 1

Scaling

- In order to add another dimension of dynamic control, consider adding a scaling factor to the input wave

$$F(ax) = d_0 + d_1 ax + d_2 a^2 x^2 + \cdots + d_N a^N x^N$$

- This parameter is called the distortion index
  - Varies between 0 and 1
  - Increases the harmonic complexity
- Example: $F(x)=x + x^3 + x^5$
  - $h_1(a)=a+0.75*a^3+0.625*a^5$
  - $h_3(a)=0.25*a^3+0.0625*a^5$
  - $h_5(a)=0.0625*a^5$

Polynomial Selection

- Spectral matching uses specific polynomial combinations to get the spectrum of the waveshaper output to match a desired spectrum
- Use Chebyshev Polynomials because of their well documented behavior
  - For a cosine input with amplitude 1, $T_k(x)$ contains only the kth harmonic.
  - Can add multiple Chebyshev Polynomials to get exact like the desired transfer function.
  - For a<1 the outputs properties do not hold

Chebyshev Polynomials

$$T_0(x)=1$$
$$T_1(x)=x$$
$$T_2(x)=2x^2-1$$
$$T_3(x)=4x^3-3x$$
$$T_4(x)=8x^4-8x^2+1$$
$$T_5(x)=16x^5-20x^3+5x$$
$$T_6(x)=32x^6-48x^4+18x^2-1$$

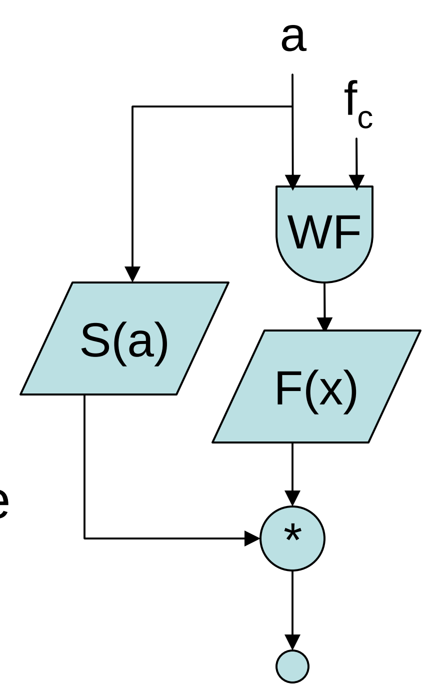
$$T_{k+1}(x)=2xT_k(x)-T_{k-1}(x)$$

Dynamic Properties

- While the complexity evolves from $0 < a < 1$, the harmonics do not change monotonically.
  - Even if final waveform does not have many upper harmonics, the ripples in the 0 to 1 range may create a brassy sound before the desired spectrum
  - The higher the order, the harder the problem.
- Introducing sign flips increases the smoothness as the spectrum evolves
  - The even harmonics should have a +,-,+,-,+,-… pattern starting at the zeroth harmonic
  - The odd harmonics should have a +,-,+,-,+,-… starting with the first harmonic
  - Combined even and odd will have a +,+,-,-,+,+,-,-,… pattern
  - Examples in Figure 5.26 from Jerse.

Implementation

- Instead of direct evaluation, transfer functions are implemented as look up tables.
- Amplitude Scaling
  - Since the amplitude of the input sine wave affects the spectral content, can be good to add extra blocks to use it to control spectrum and overall amplitude.
  - Use an extra scaling function while controls the relationship between the richness and the output loudness
  - Often used to keep the output power constant with different spectral shapes.



References

1. Dodge, C. & Jerse T. Computer Music, Schirmer Books, NY, 1997

Csound w/ GUI web page

- http://music.calarts.edu/~bcassidy/CompMusPC/
  - Lots of C sound links and a MIDI to Csound converter
- http://en.wikipedia.org/wiki/Frequency_modulation_synthesis
- http://ccrma.stanford.edu/software/snd/snd/fm.html
  - Good technical discussion of FM