CS 583– Computational Audio -- Fall, 2021

Wayne Snyder
Computer Science Department
Boston University

Lecture 5

Conclusions on Ring and Frequency Modulation:

- What happens to the spectra?
- Frequency Modulation with varying parameters

Physical Modeling Synthesis:

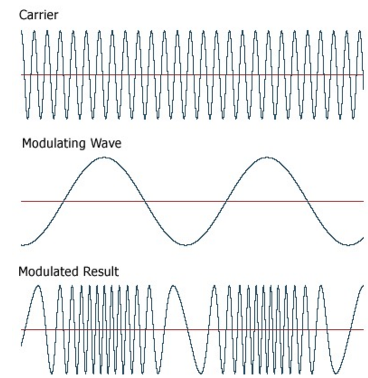
- The Karplus-Strong String Synthesis Algorithm

Computer Science

Frequency Modulation

There is a little problem with a straight-forward implementation of frequency modulation; here is a naïve way of changing the frequency of the audio signal:

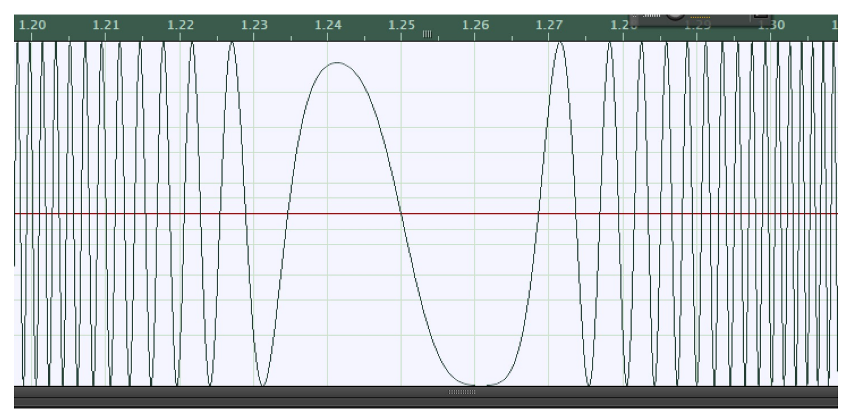
```python
# Take a signal of freq f1 and amplitude A1 and modify it using freq modulation
# of freq f2 and amplitude A2

def freqModulationNaive(f1,A1,f2,A2,duration):
    X = [0] * (SR * duration)
    for k in range( SR*duration ):
        freqIncr = A2 * np.sin(2*np.pi*f2*k/SR)
        X[k] = A1*MAX_AMP*np.sin(2*np.pi*(f1+freqIncr)*k/SR)
    return X
```



So you would expect that

X = freqModulationNaive(440,1.0,6,10,5)

would vary the 440 audio frequency by 10 Hz at the rate of 6 Hz, oscillating between 430 Hz and 450 Hz 6 times a second. But here is what you get:



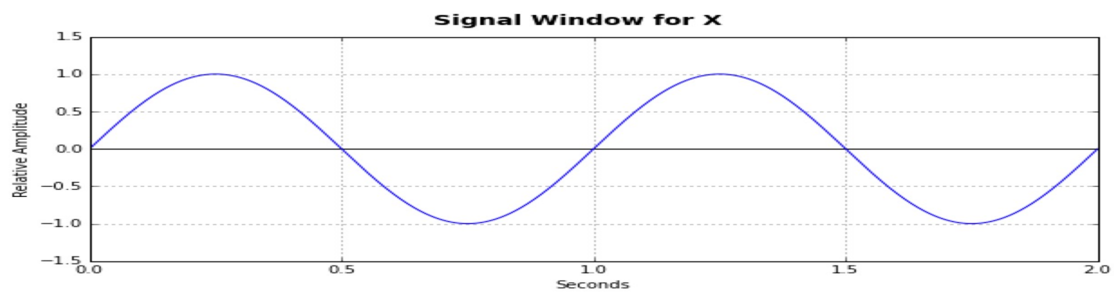
Frequency Modulation

Computer Science

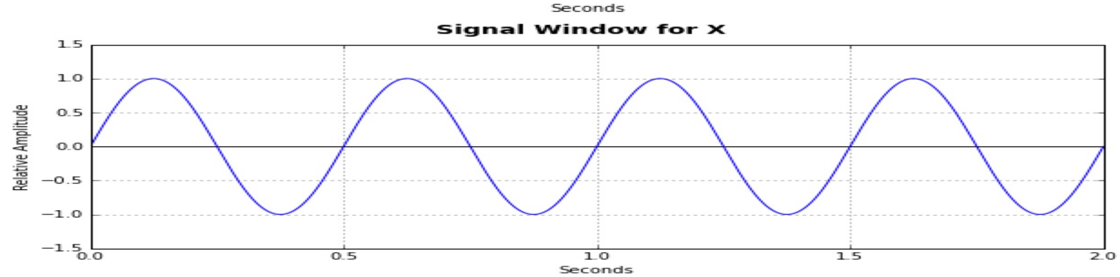
What is the problem? When you change the frequency, but only keep track of time since the beginning of the signal, the (instantaneous) phases may not match.

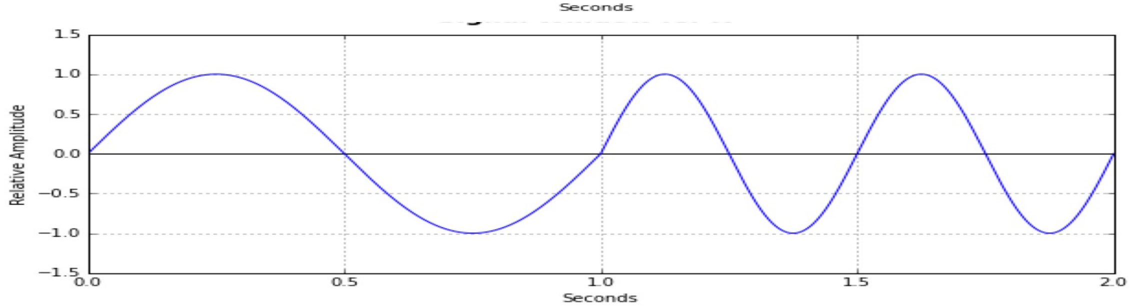
Let's consider what happens when you change a signal suddenly between two frequencies, say 1 Hz and 2 Hz. If we have a 2 sec signal, and change at the 1 sec mark, we get a (somewhat) smooth transition, because the (instantaneous) phase at the transition point was the same:

$\sin(\ 2 * \pi * 1 * t\ )$



$\sin(\ 2 * \pi * 2 * t\ )$

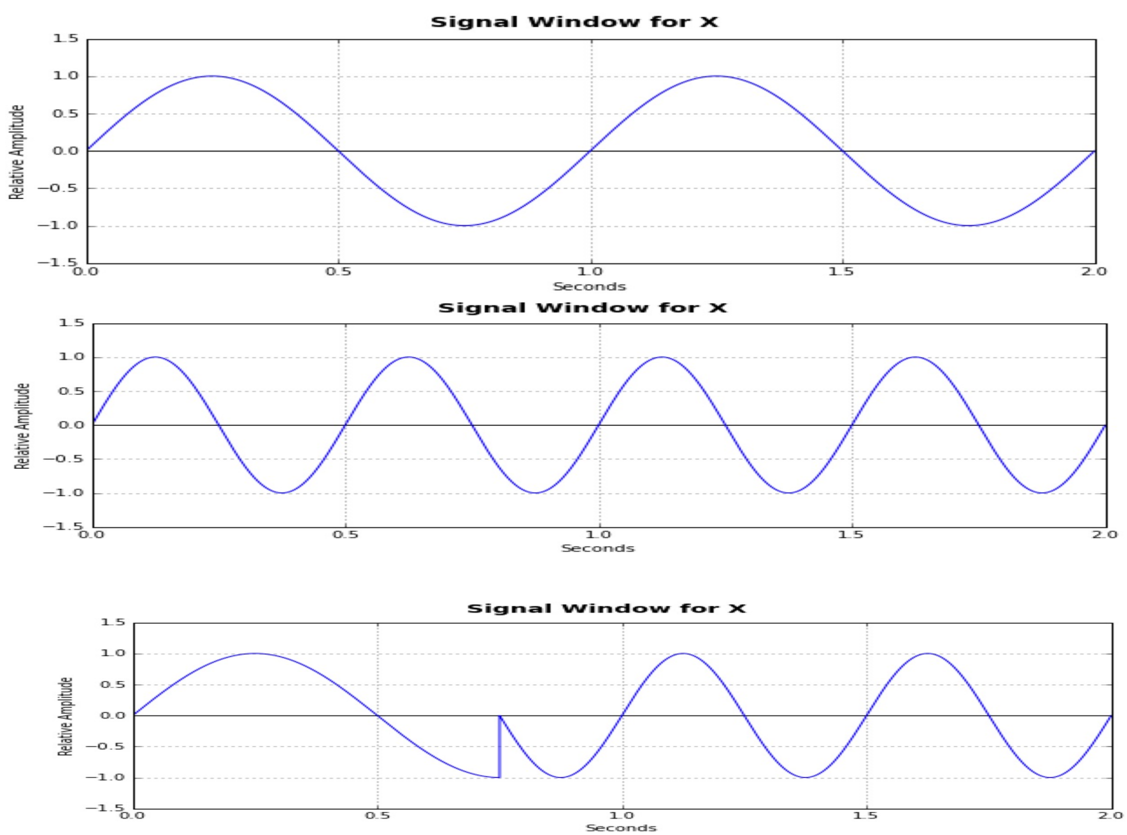




Frequency Modulation

What is the problem? When you change the frequency, but only keep track of time since the beginning of the signal, the (instantaneous) phases may not match.

If we change the frequency at 0.75 sec, however, **the (instantaneous) phases do not match**, and we get a discontinuity in the signal:



Frequency Modulation

Here is the corrected function:

```
def freqModulation(f1,A1,f2,A2,duration):
    X = [0] * (SR * duration)
    phase = 0.0
    newFreq = f1
    for k in range( SR*duration ):
        freqIncr = A2 * np.sin(2*np.pi*f2*k/SR)  # modulating signal
        oldFreq = newFreq
        newFreq = f1 + freqIncr
        phase +=  2 * pi * (k / SR) * (oldFreq - newFreq)
        X[k] = A1*MAX_AMP*np.sin(2*np.pi*newFreq*k/SR + phase)
    return X
```

Note that the phase has to be updated every time through the loop, to keep a running count of how far the phase has shifted each time the frequency is changed!

AM, FM, and Spectra

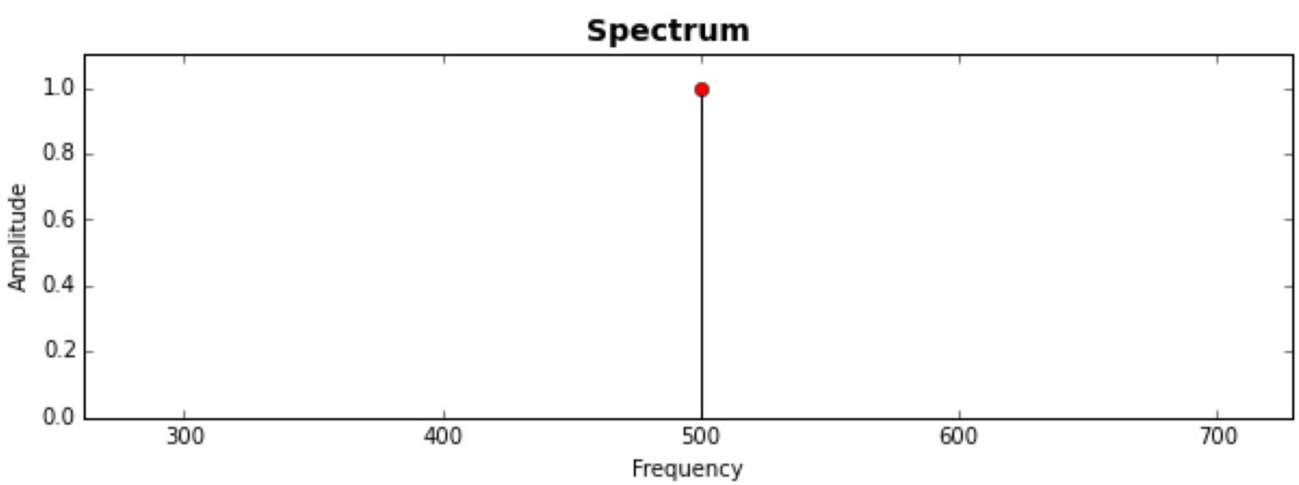
When the modulating frequency is below 20 Hz, you will hear the individual modulations:

AM & Ring Modulation = Tremolo

Frequency Modulation = Vibrato

However, when the modulating frequency is above 20 Hz, it affects the timbre of the sound, and this can be understood in terms of its spectrum….

Here is a simple 500 Hz signal and its spectrum:

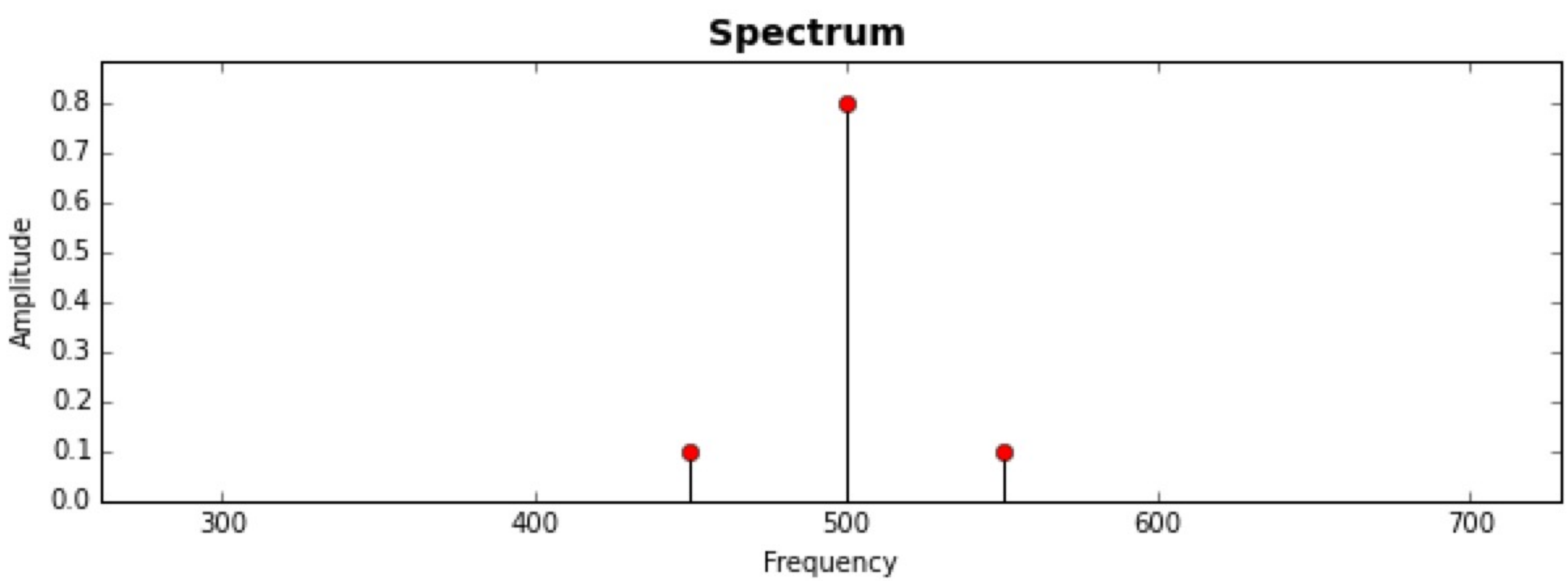


AM, FM, and Spectra

Computer Science

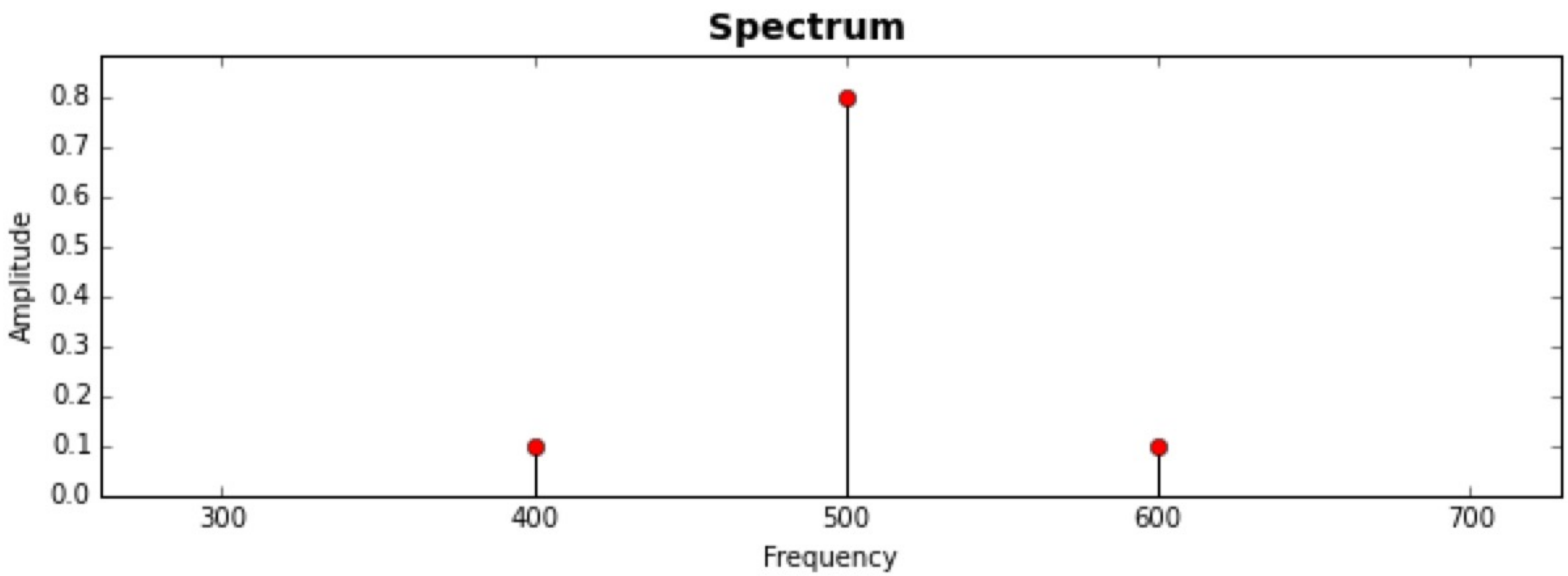
Let us first consider Amplitude Modulation: Here is the spectral analysis for a 500 Hz audio signal with AM applied with 50 Hz and 0.2 amplitude:

```
Carrier Freq: 500
Modulator Freq: 50
Modulator Amp: 0.2
tremolo.wav written.
```



AM, FM, and Spectra

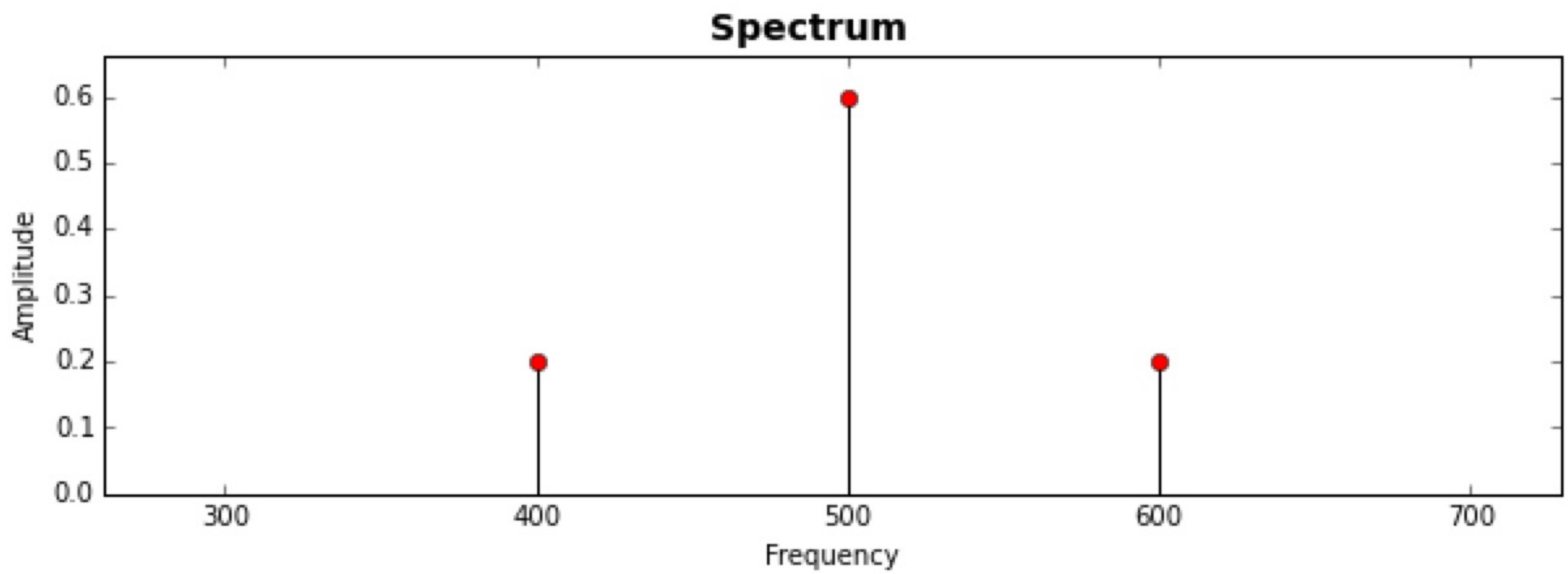

```
Carrier Freq: 500
Modulator Freq: 100
Modulator Amp: 0.2
tremolo.wav written.
```



AM, FM, and Spectra

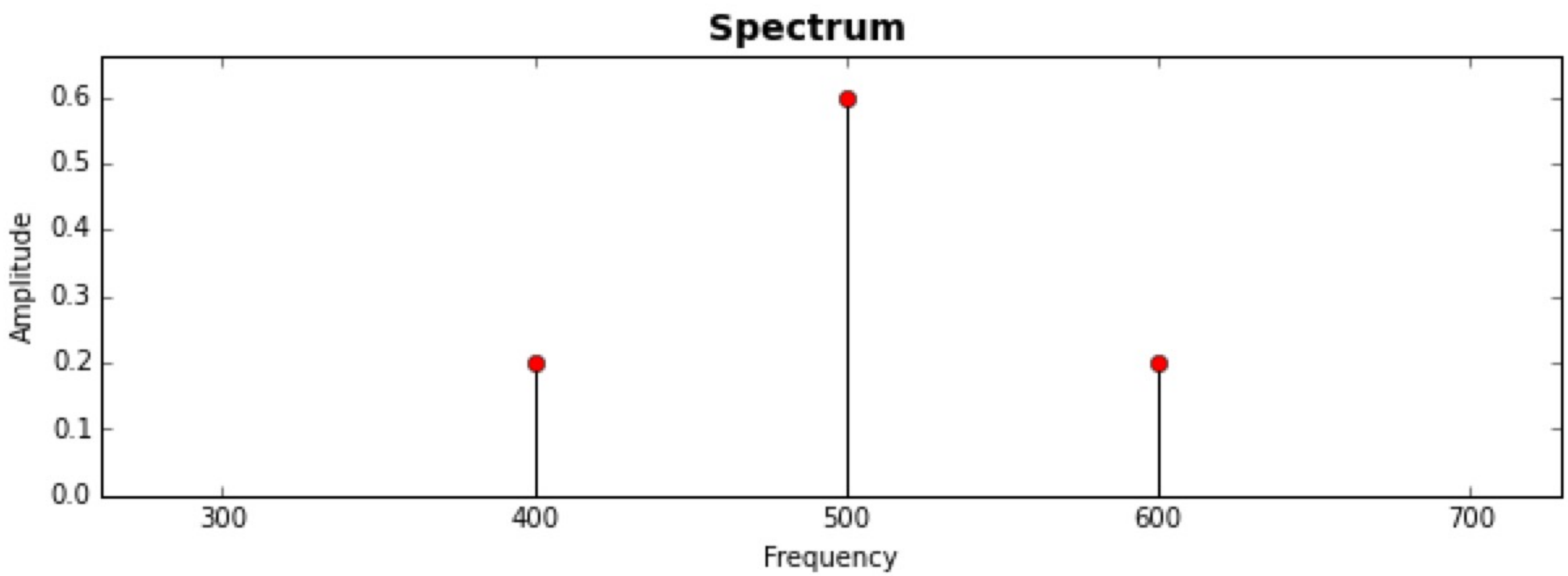
Computer Science

```
Carrier Freq: 500
Modulator Freq: 100
Modulator Amp: 0.4
tremolo.wav written.
```



AM, FM, and Spectra

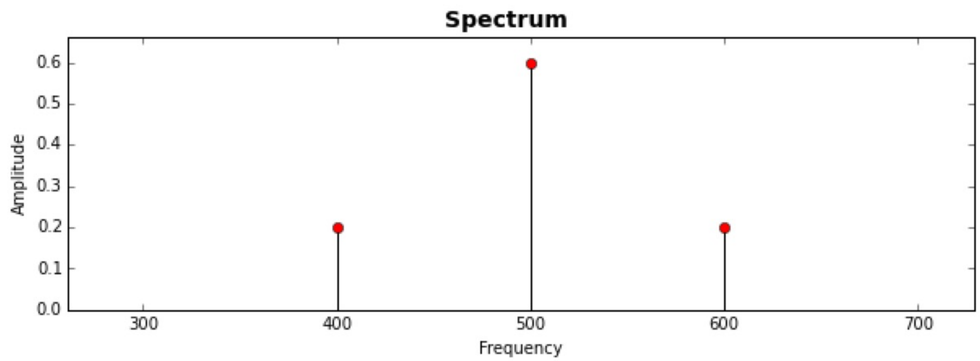

```
Carrier Freq: 500
Modulator Freq: 100
Modulator Amp: 0.4
tremolo.wav written.
```



So what is the rule for predicting the spectrum from the Amplitude Modulation frequencies and amplitude?

AM, FM, and Spectra


```
Carrier Freq: 500
Modulator Freq: 100
Modulator Amp: 0.4
tremolo.wav written.
```



So what is the rule for predicting the spectrum from the Amplitude Modulation frequencies and amplitude?

f_c at amplitude $A_c - A_m$
$f_c - f_m$ at amplitude $A_m/2$
$f_c + f_m$ at amplitude $A_m/2$

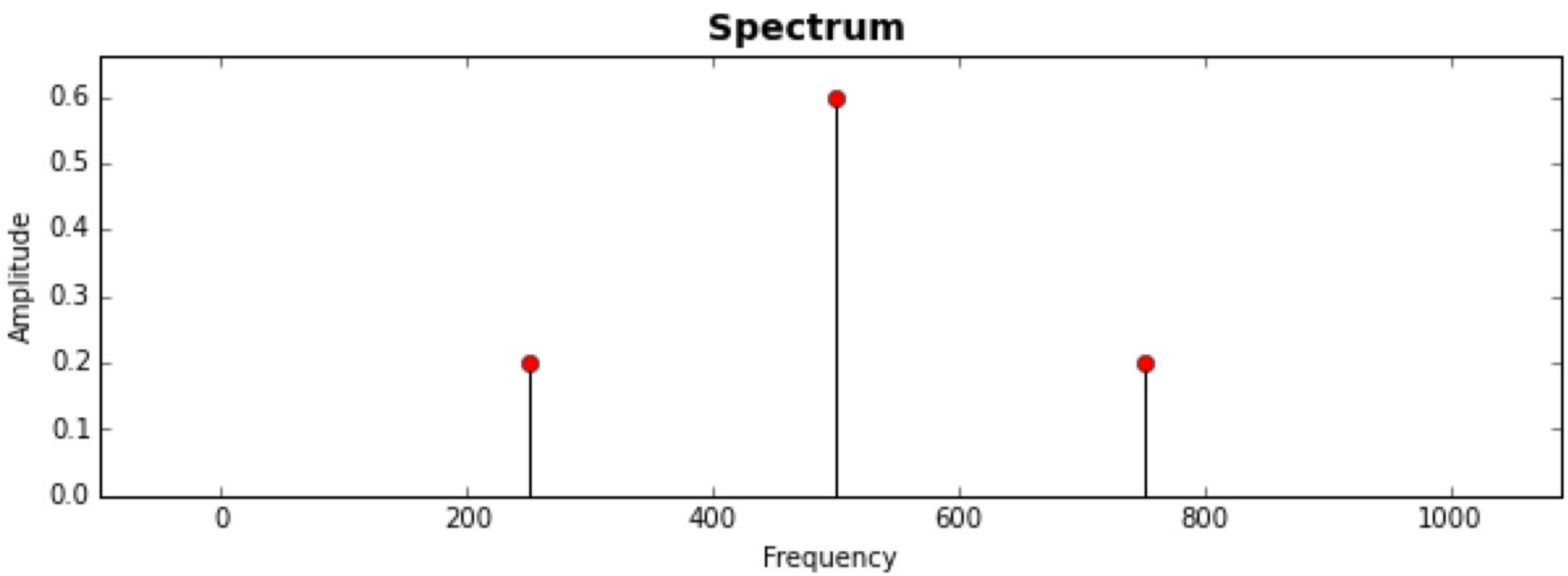
f_c = Carrier Frequency
A_c = Carrier Amplitude
f_m = Modulator Frequency
A_m = Modulator Amplitude

The frequencies on each side of the carrier frequency are called sideband frequencies.

AM, FM, and Spectra

Note that if you choose your frequencies carefully, you can create a harmonic series, which will sound reasonably pleasant:

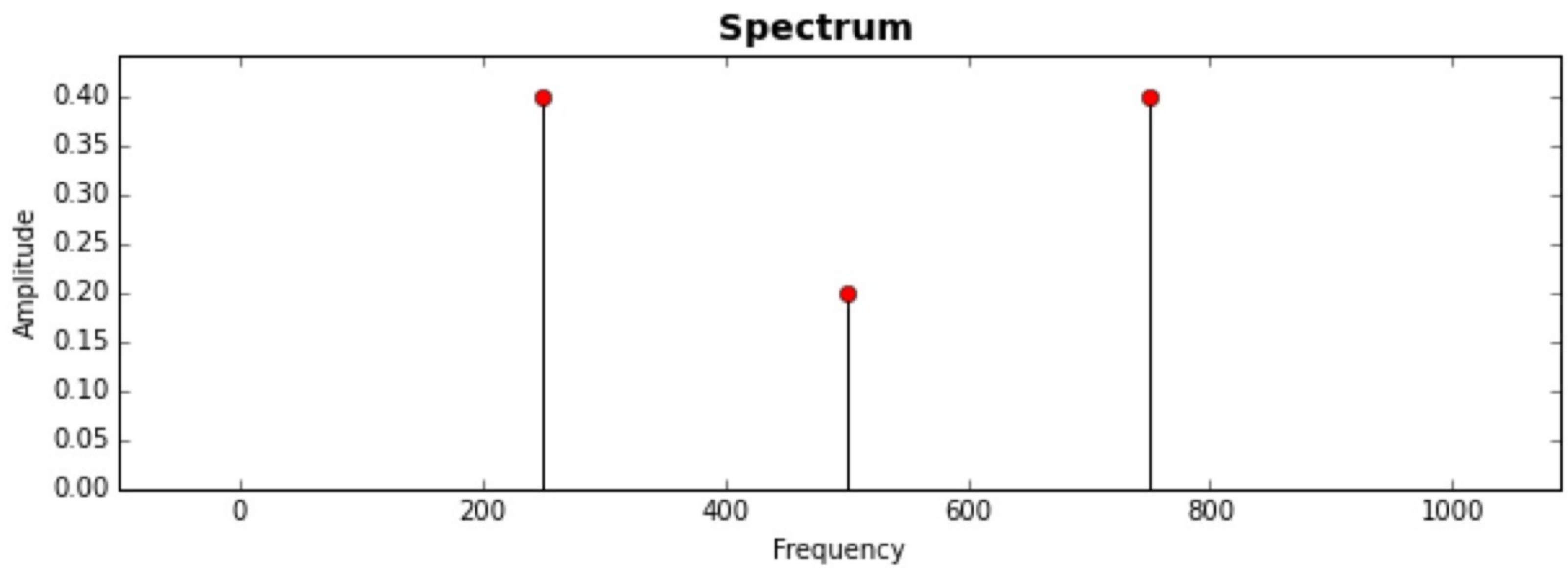
```
Carrier Freq: 500
Modulator Freq: 250
Modulator Amp: 0.4
tremolo.wav written.
```



AM, FM, and Spectra

Note that if you choose your frequencies carefully, you can create a harmonic series, which will sound reasonably pleasant:

```
Carrier Freq: 500
Modulator Freq: 250
Modulator Amp: 0.8
tremolo.wav written.
```

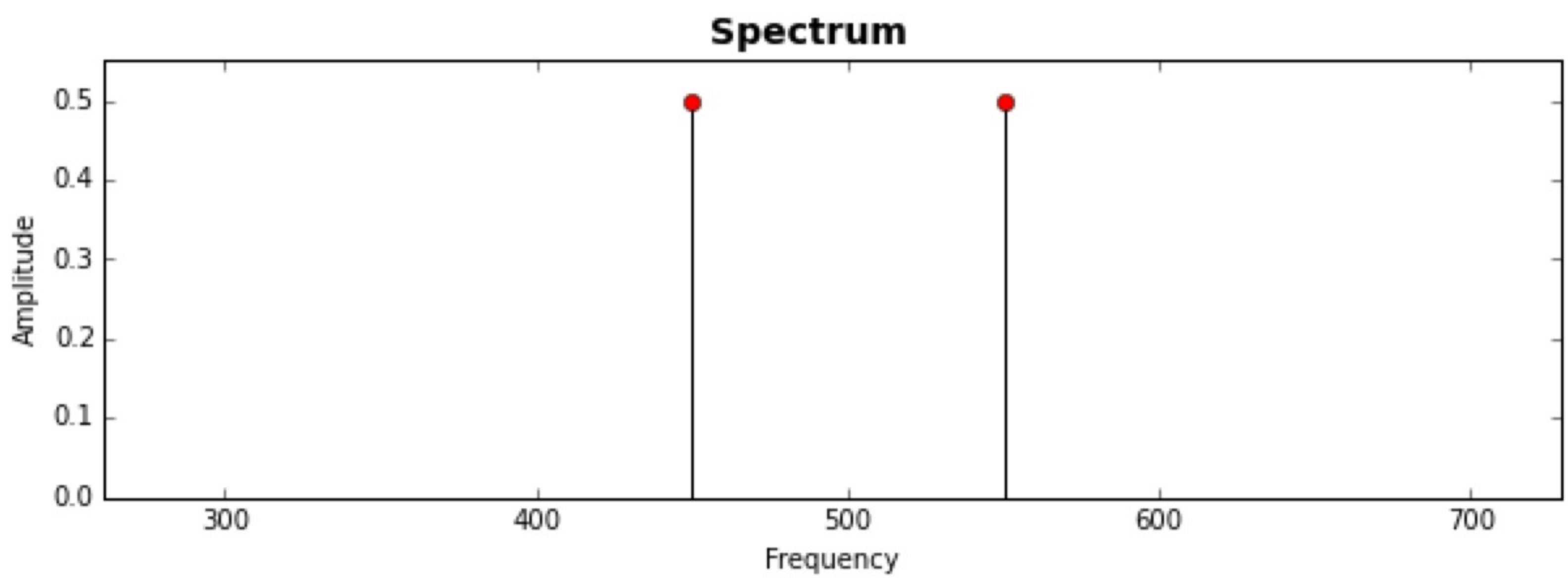


AM, FM, and Spectra

Computer Science

Now let us try Ring Modulation: Here is the spectral analysis for a 500 Hz audio signal with a Ring Modulation at 50 Hz:

```
Carrier Freq: 500
Modulator Freq: 50
ring.wav written.
```



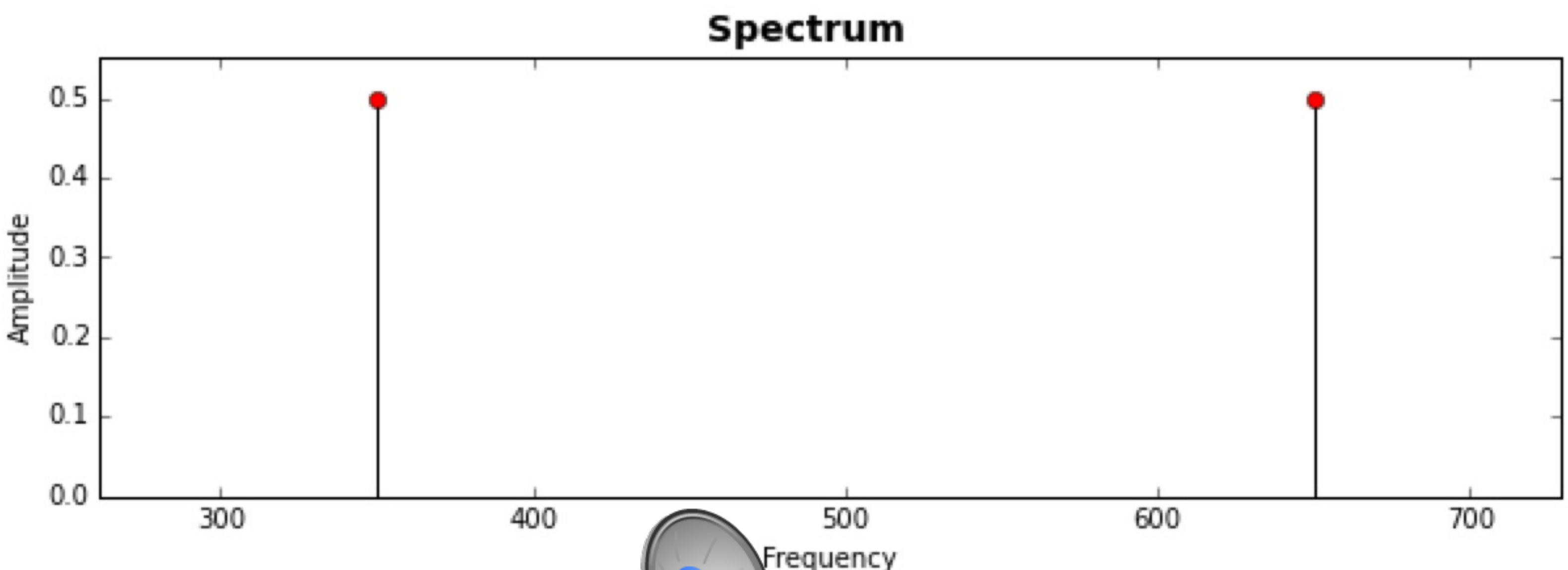
AM, FM, and Spectra

Computer Science

Here is another one. What is the rule??

Carrier frequency disappears and is replaced by sideband frequencies, which divide the amplitude between them!

```
Carrier Freq: 500
Modulator Freq: 150
ring.wav written.
```



AM, FM, and Spectra

Computer Science

Rule for spectrum in Ring Modulation with (f_c, A_c, _) * (f_m, A_m, _)

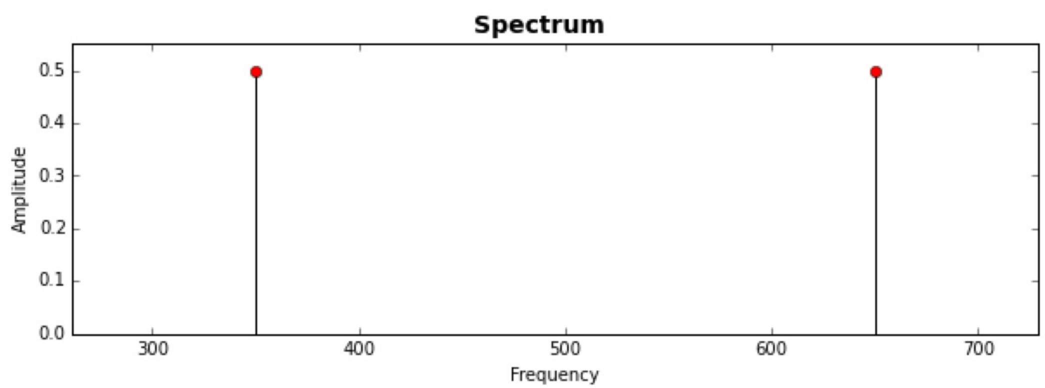
(f_c, A_c, _) disappears, replaced by
$f_c - f_m$ at amplitude $(A_c+A_m)/2$
$f_c + f_m$ at amplitude $(A_c+A_m)/2$

f_c = Carrier Frequency
A_c = Carrier Amplitude
f_m = Modulator Frequency
A_m = Modulator Amplitude

This is consistent with the trig formula for the product of sin waves:

$$\sin(x)*\sin(y) = \tfrac{1}{2} [\cos(x-y) - \cos(x+y)]$$

```
Carrier Freq: 500
Modulator Freq: 150
ring.wav written.
```



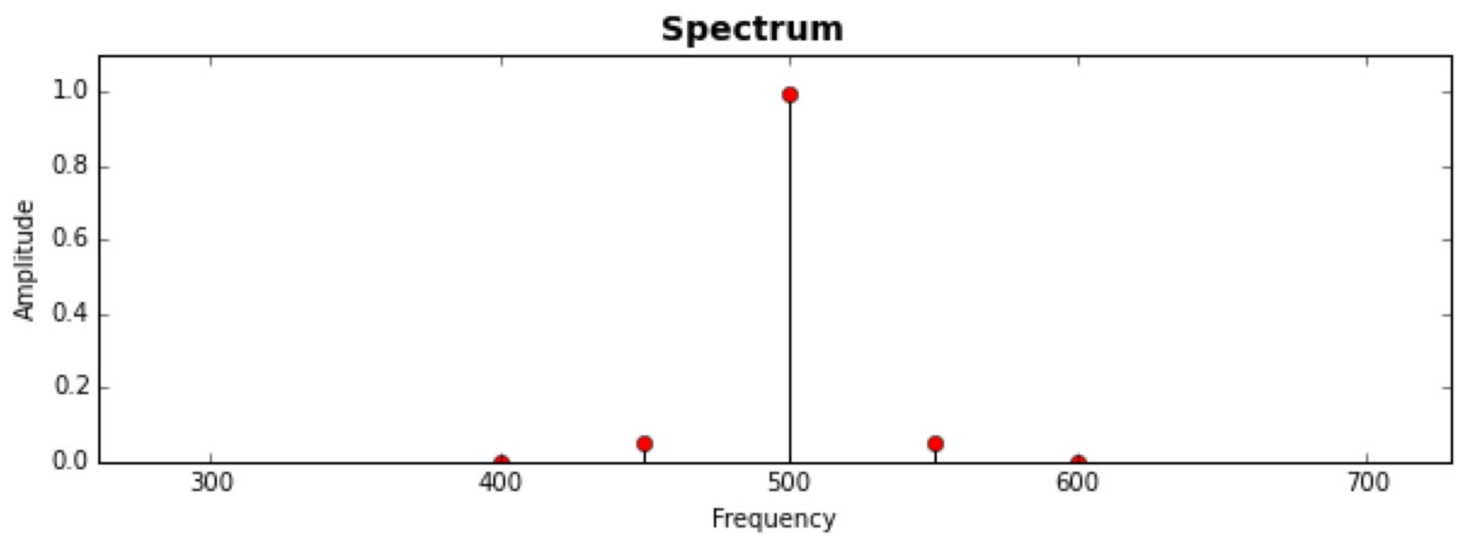
AM, FM, and Spectra

Now Frequency Modulation! Let's try the same thing….

The Modulation Index is the ratio A_m / f_m

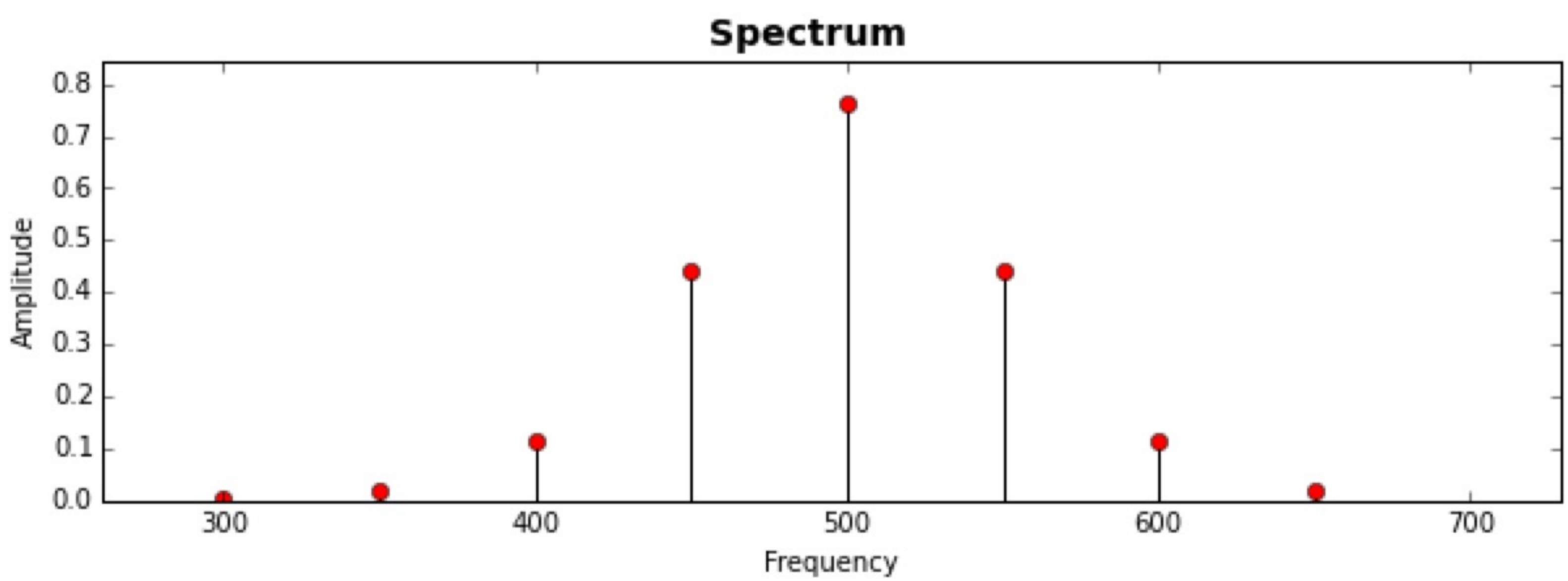
So here the carrier frequency varies between 495 .. 505 Hz.

```
Carrier Freq: 500
Modulator Freq: 50
Modulation Index: 0.1
fm.wav written.
```



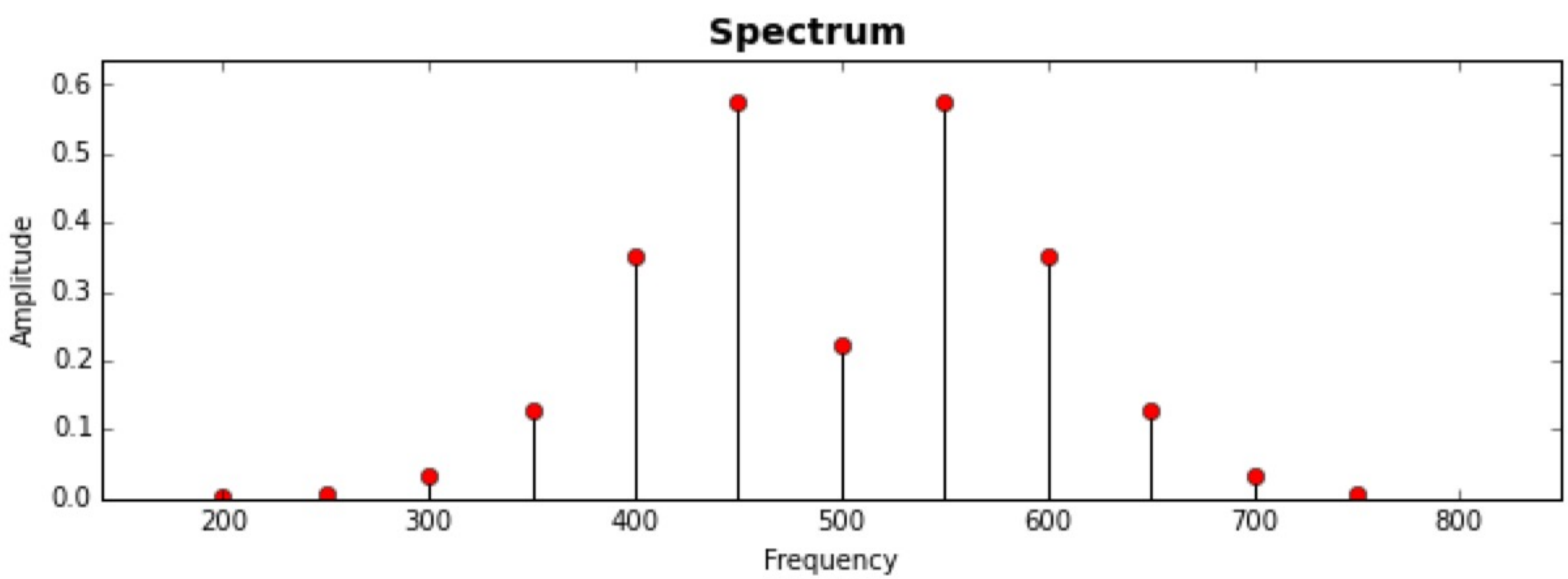
AM, FM, and Spectra


```
Carrier Freq: 500
Modulator Freq: 50
Modulation Index: 1.0
fm.wav written.
```



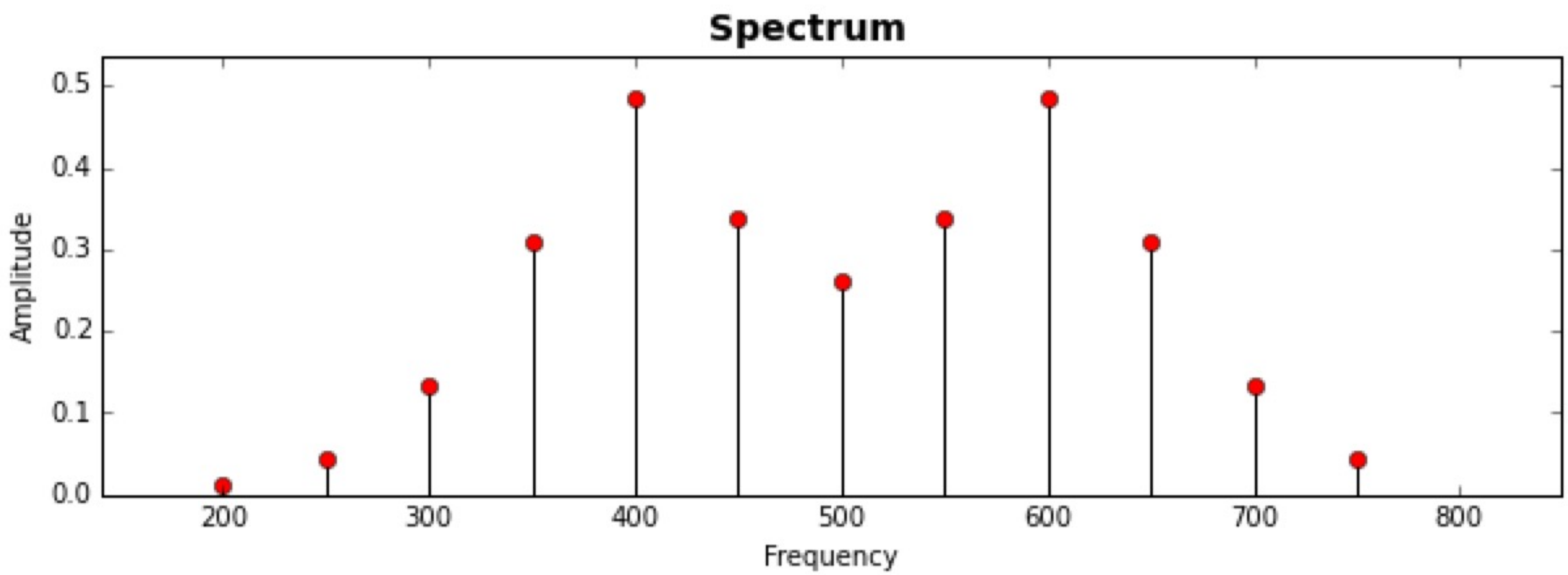
AM, FM, and Spectra


```
Carrier Freq: 500
Modulator Freq: 50
Modulation Index: 2.0
fm.wav written.
```



AM, FM, and Spectra

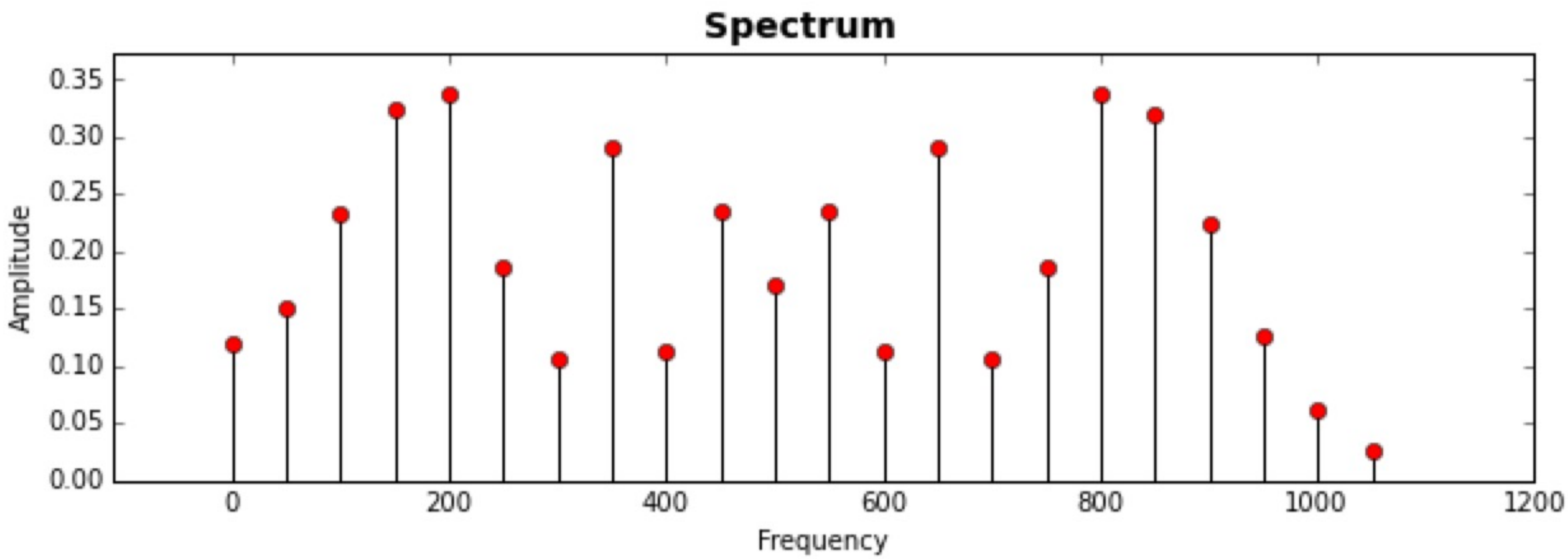

```
Carrier Freq: 500
Modulator Freq: 50
Modulation Index: 3.0
fm.wav written.
```



The “Rule” is a little hard to see, but can be mathematically described; roughly, you get about 2*Index sideband frequencies on each side, at intervals of the modulator frequency.

AM, FM, and Spectra

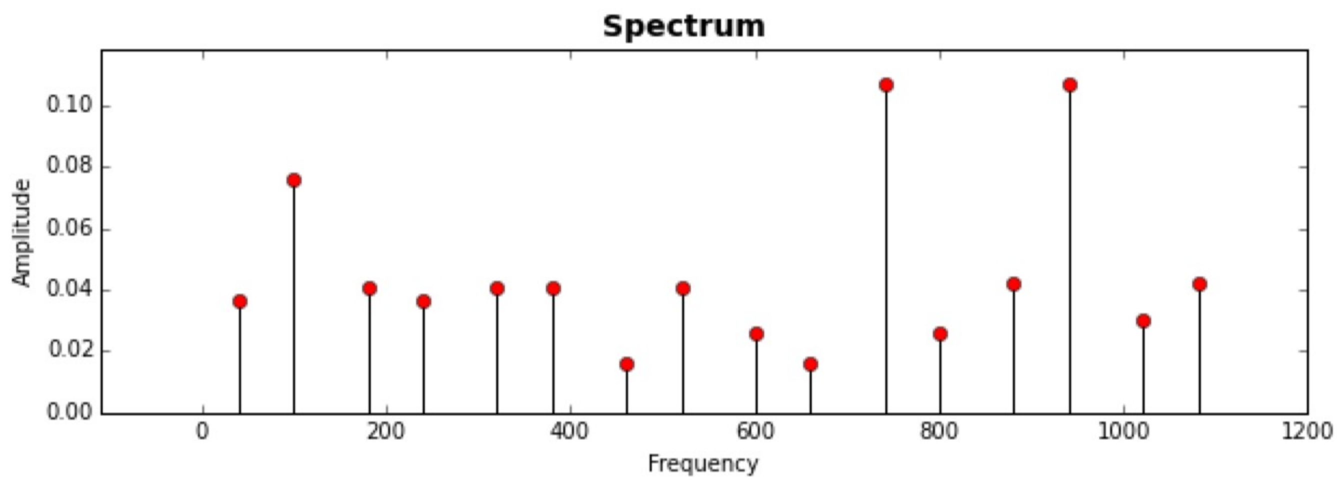

```
Carrier Freq: 500
Modulator Freq: 50
Modulation Index: 8.0
fm.wav written.
```



Time-Varying Spectra and FM

This led researchers (especially John Chowning) to try to design instrumental sounds by setting the parameters to extreme values to get interesting spectra. Here the modulating frequency is LARGER than the carrier frequency, and varies so that the frequency swings between positive and negative values!

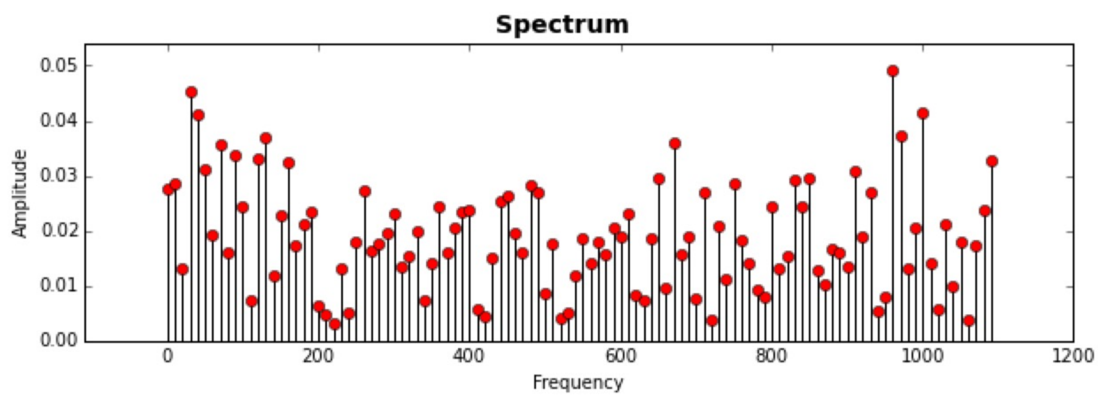
```
Carrier Freq: 100
Modulator Freq: 280
Modulation Index: 6
fm.wav written.
```



Time-Varying Spectra and FM

Not a great sound! But when combined with amplitude envelopes and by varying the parameters to simulate the way that spectra change as high frequencies “roll off” over time, you can get interesting simulations! One common technique is to decrease the modulation index over time:

```
Carrier Freq: 100
Modulator Freq: 280
Modulation Index: 6.0-> 0.0
fm.wav written.
```



To a notebook to hear various examples of varying modulation indices……

http://cmc.music.columbia.edu/musicandcomputers/chapter4/04_07.php

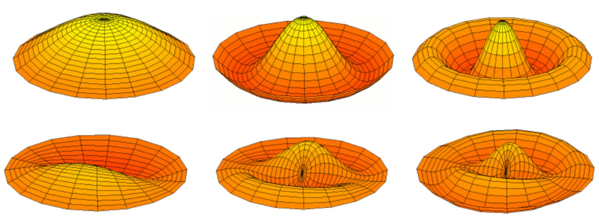
Physical-Modelling Synthesis

Computer Science

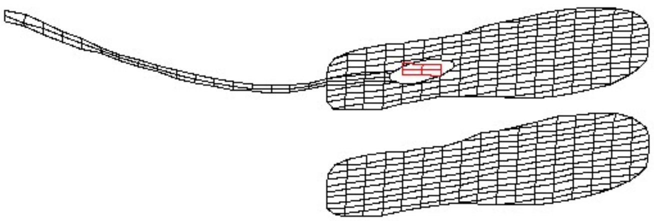
Physical modelling synthesis refers to sound synthesis methods in which the waveform of the sound to be generated is computed using a mathematical model, a set of equations and algorithms to simulate a physical source of sound, usually a musical instrument.

Modelling attempts to replicate laws of physics that govern sound production, and will typically have several parameters, some of which are constants that describe the physical materials and dimensions of the instrument, while others are time-dependent functions describing the player's interaction with the instrument, such as plucking a string, or covering toneholes.

-Wikipedia

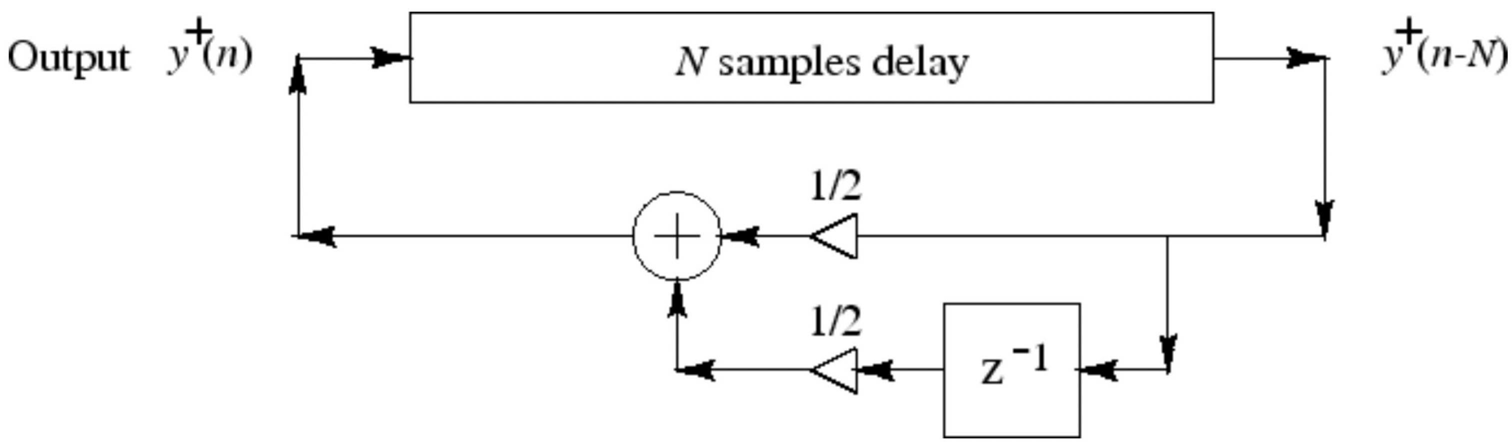


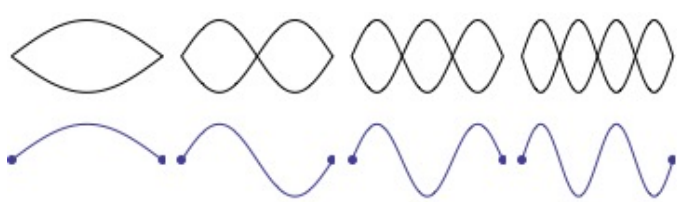
Drumhead Vibration Modes



Vibrating String Synthesis: Karplus-Strong Algorithm

An elegant model of a vibrating string (or a vibrating column of air, or …) is the **Karplus-Strong Algorithm**, which uses a queue to hold samples; the queue is filled with random values, then "rotated" and a filter applied to new samples inserted; the simplest filter is an averaging filter:





Vibrating String Synthesis: Karplus-Strong Algorithm

To a Jupyter notebook….

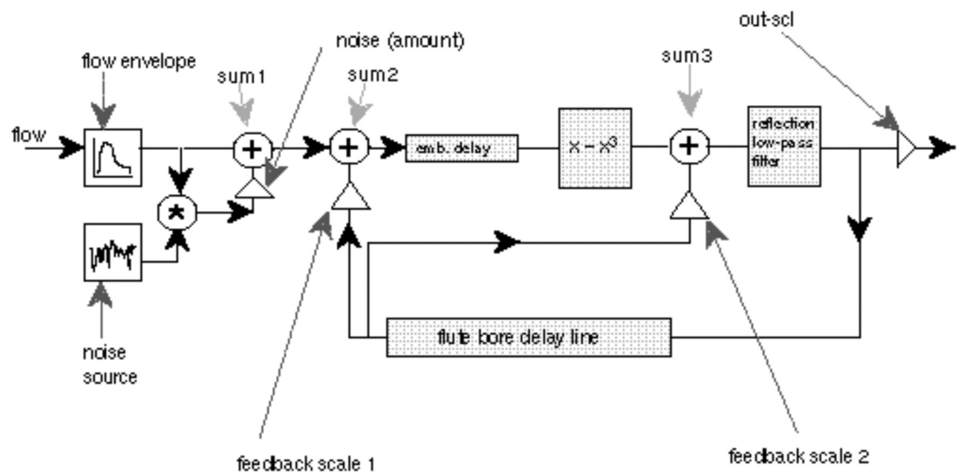
Additional Examples may be found here:

https://ccrma.stanford.edu/~jos/Mohonk05/Karplus_Strong_Algorithm.html

Vibrating String Synthesis: Perry Cooks Slide Flute

Computer Science

This simple and effective technique has been extended many different directions, to specify particular instruments, such as Perry Cook's Slide Flute:



Various examples of physical modelling synthesis may be found here:

https://www.dsprelated.com/freebooks/pasp/Sound_Examples.html