Perception and Multimedia Computing

Sound Waves: Equations and Audio

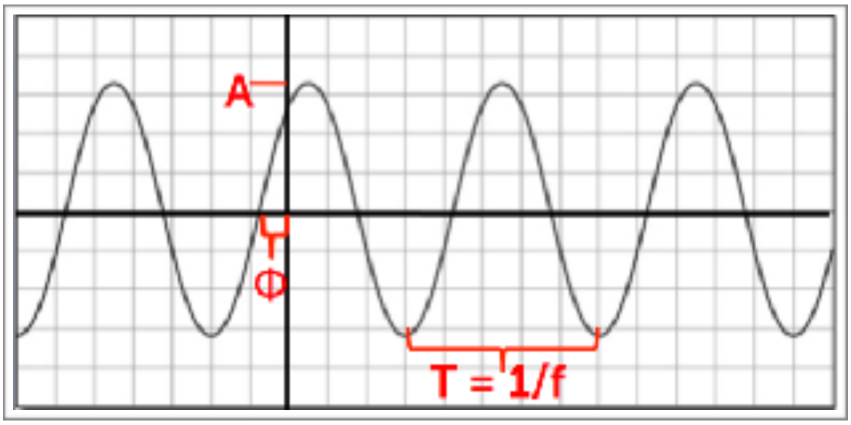
Friday 8th Dec 2017

1. Understanding the sine equation

Recall that the generic equation for a sine wave at time *t* is:

$$f(t) = A \sin(2\pi f t + \varphi)$$

where *A* is amplitude, *f* is frequency (which is 1/*T,* where *T* is the period), and Φ is phase offset (expressed as a value between 0 and 2π):

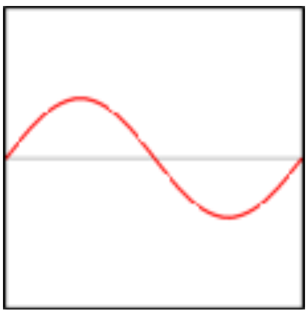


Derive the equations for the three sine waves below. (That is, figure out what *A*, *f*, and Φ should be for each one).

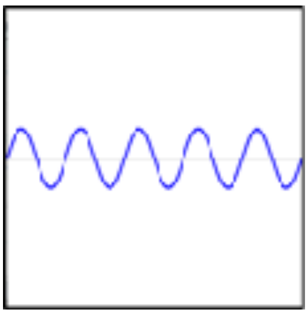
Then edit the **p5.js skeleton code for this lab (download from learn.gold)** to include these equations, so that it draws all three waves:

a. A sine whose peaks are -100 and 100, which completes 1 cycle in 500 pixels (i.e., its period is 500 pixels).
b. A sine whose peaks are -50 and 50, which completes 5 cycles in 500 pixels.
c. A sine whose peaks are -200 and 200, which completes 3 cycles in 500 pixels, and which starts at 0 going downwards instead of starting at 0 going upwards.

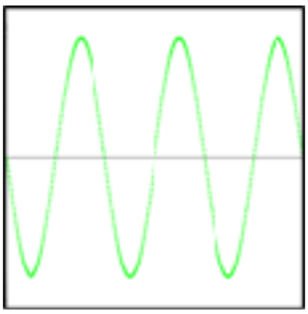
Your sketch should look like (d), and you should have a good understanding of how *A*, *f*, *T*, and Φ influence the wave:



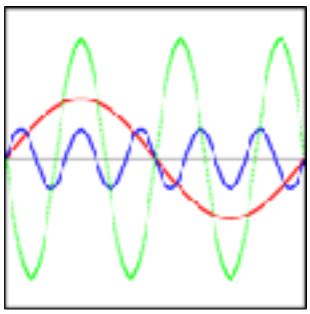
(a)



(b)



(c)



(d)

2. Experimenting with sound:

The two demo sketches provided allow you to change the frequency and amplitude of a sine wave with your mouse or by typing in numbers. Use them to experiment with changing the frequency and amplitude of a sine wave. Investigate and answer the following questions.

a. What is the highest frequency you can hear?

b. Does this change when you are using headphones versus the built-in computer speakers?

c. When you keep amplitude of the sine wave constant, what is the loudest frequency for you when you are using headphones?

d. What is the loudest frequency when you are using the built-in computer speakers? Is it the same as when you are using headphones? Why do you think this is the case?

e. Using the SineWaveKeyboard example, listen to the difference between a sine at 400 Hz and a sine at 500Hz. (In music, this difference is called an “interval.”) Now, listen to the difference between a sine at 1600 Hz and a sine at 1700 Hz. Does the 1600/1700Hz pair sound closer together or further apart than the 400/500Hz pair?

f. 500 Hz is exactly 100 Hz above 400 Hz, and 1700 Hz is exactly 100 Hz above 1600 Hz. So why don’t these two intervals sound the same?

g. What frequency would you pair with 1600Hz to get the same interval (perceptual distance) as the 400/500Hz pair?

3. Adding sines together

Start with either of the sine wave example sketches from q2 and modify it, so that it plays multiple sine waves simultaneously (using multiple maxiOsc objects).

a. Play multiple sine waves that are harmonically related—that is, whose frequencies are all integer multiples of a single *fundamental* frequency. For instance, if your fundamental is 400 Hz, add together sines of 400Hz, 2*400=800 Hz, 3*400=1200Hz, etc. Note: be careful to scale the amplitude so that the maximum value of your sum is never greater than 1! The easiest way to do this is to use the `.amp()` function to set the gain of each maxiOsc to 1.0/*N*, where *N* is the number of sines you’re playing. Now answer the following questions:

- What happens to the *pitch* of the sound as you add more frequencies?

- What happens to the sound colour / timbre / tone quality as you add more frequencies?

b. Make a copy of your sketch and change it so that you instead play sine waves of different frequencies which are *not* harmonically related.

- What do you hear as you add more frequencies? How is this different from when you added frequencies that were harmonically related?

- How does the sound change when the range of frequencies is very big (e.g., a spread of thousands of Hertz) or very small (e.g., all within a few Hertz of each other)?

- For extra fun, try using an array of maxiOsc objects so that you can easily add many (dozens!) of sines together at *random* frequencies. This will create some interesting audio effects…