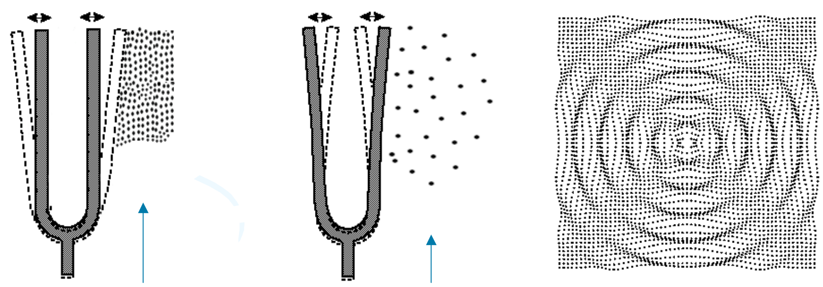
What is Sound?

As the tines move back and forth they exert pressure on the air around them.
(a) The first displacement of the tine compresses the air molecules causing high pressure.
(b) Equal displacement of the tine in the opposite direction forces the molecules to widely disperse themselves and so, causes low pressure.
(c) These rapid variations in pressure over time form a pattern which propogates itself through the air as a wave. Points of high and low pressure are sometimes reffered to as '**compression**' and '**rarefaction**' respectively.

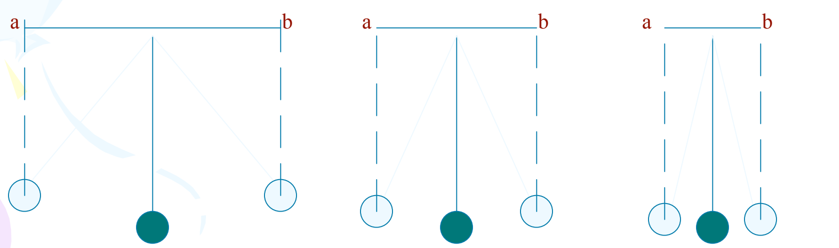


(a) compression (b) rarefaction (c) wave propagation of a tuning fork as seen from above

Simple Harmonic Motion -- a Pendulum

- When a pendulum approaches equlibrium it doesn't slow down; it simply travels a smaller distance from the point of rest.
- Any body undergoing simple harmonic motion moves periodically with uniform speed.
- If the tuning fork is moving periodically then the pressure variations it creates will also be periodic.

The time taken to get from position a to b in all three cases is the same



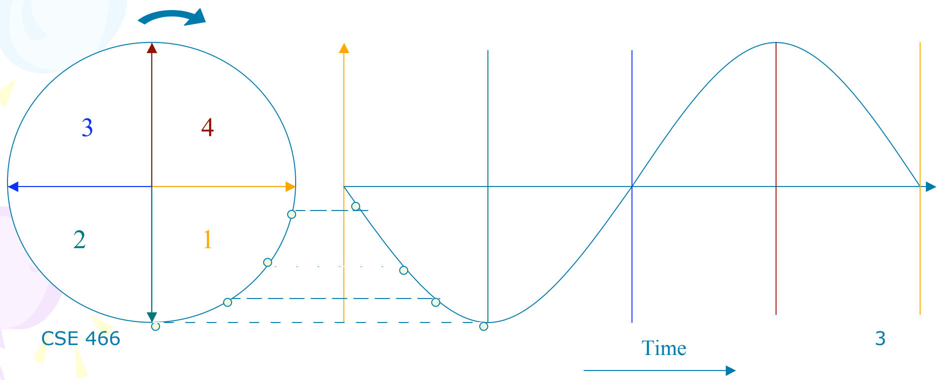
Maximum displacement at 0 seconds | Maximum displacement after say, 3 seconds | Maximum displacement after say, 6 seconds

The Unit Circle

These pressure patterns can be represented using as a circle.
Imagine the journey of the pendulum or the tine in four stages:

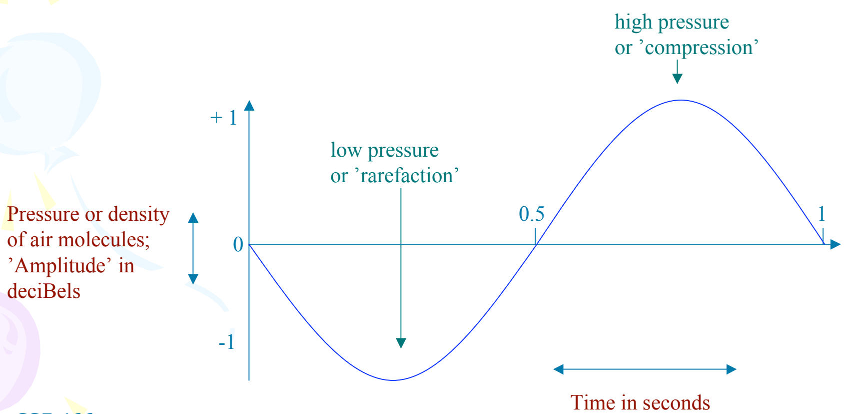
1) from its point of rest to its first point of maximum displacement...
2) its first point of maximum displacement back through the point of rest...
3) ... to its second point of maximum displacement...
4) ... and back from there through its point of rest again

We can map that journey to a circle. This is called the **Unit Circle**. The **sine wave** represents this journey around and around the unit circle *over time*.



Sine Waves

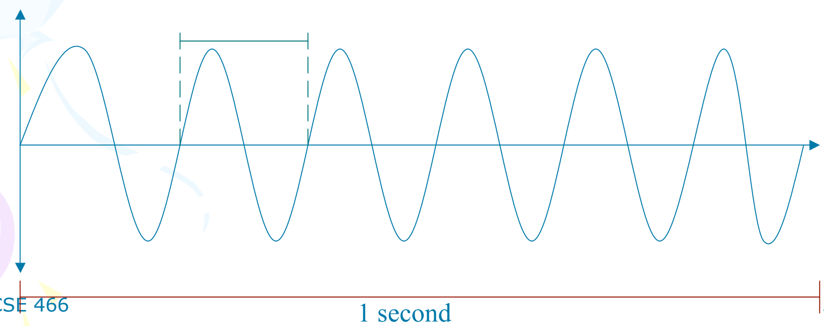
The **sine wave** or **sinusoid** or **sinusoidal signal** is probably the most commonly used graphic representation of sound waves.



Sine Waves

The specific properties of a sine wave are described as follows.

Frequency = the number of cycles per second (this wave has a frequency of 6 hertz)
Amplitude = variations in air pressure (measured in decibels)
Wavelength = physical length of 1 period of a wave (measured in metres per second)
Phase = The starting point of a wave along the y-axis (measured in degrees)



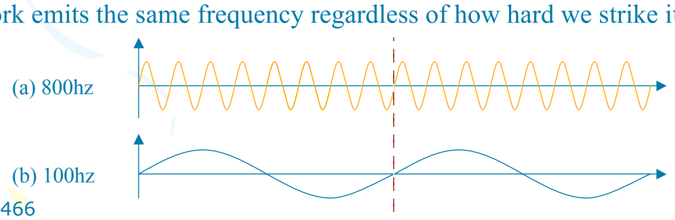
Frequency

Frequency refers to the number of cycles of a wave per second. This is measured in **Hertz**. So if a sinusoid has a frequency of 100hz then one period of that wave repeats itself every 1/100th of a second. Humans can hear frequencies between 20hz and 20,000hz (20Khz).

1) Frequency is closely related to, *but not the same as!!!*, pitch.

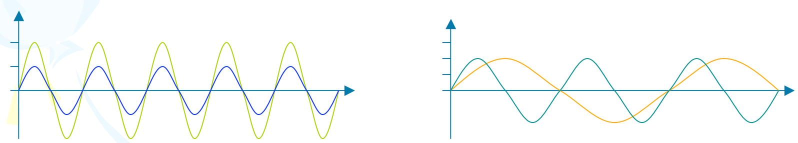
2) Frequency does not determine the speed a wave travels at. Sound waves travel at approximately 340metres/second regardless of frequency.

3) Frequency is inherent to, and determined by the vibrating body – not the amount of energy used to set that body vibrating. For example, the tuning fork emits the same frequency regardless of how hard we strike it.



Amplitude

- Amplitude describes the size of the pressure variations.
- Amplitude is measured along the vertical y-axis.
- Amplitude is closely related to *but not the same as!!!*, loudness.



(a) Two signals of equal frequency and varying amplitude

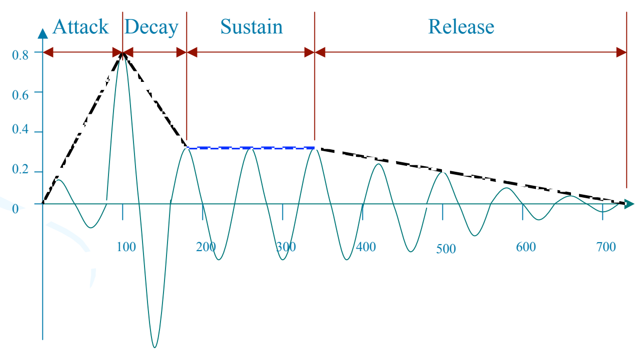
(b) Two signals of varying frequency and equal amplitude

Amplitude Envelope

The amplitude of a wave changes or 'decays' over time as it loses energy.

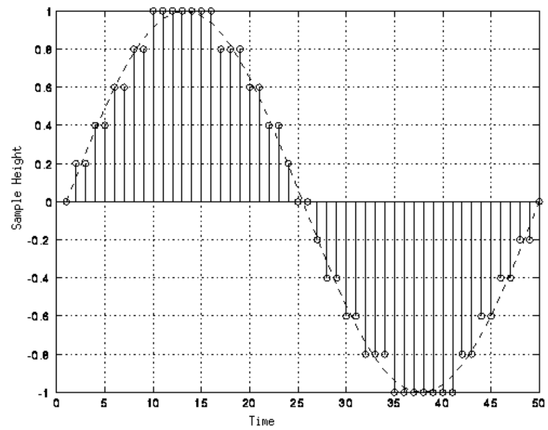
These changes are normally broken down into four stages;
Attack , **Decay**, **Sustain** and **Release**.

Collectively, the four stages are described as the **amplitude envelope**.



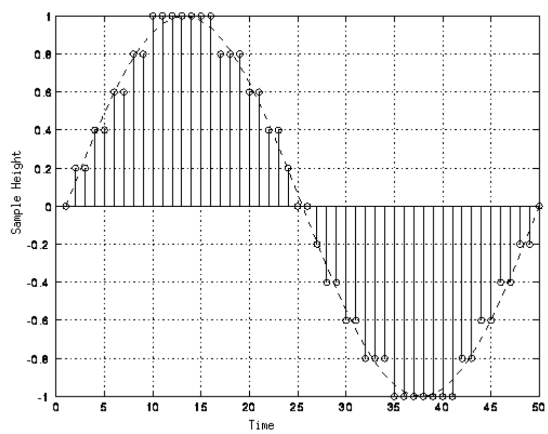
Quantization

- The digital signal is defined only at the points at which it is sampled.



Quantization

- The height of each vertical bar can take on only certain values, shown by horizontal dashed lines, which are sometimes higher and sometimes lower than the original signal, indicated by the dashed curve.



Quantization

- The difference between a quantized representation and an original analog signal is called the *quantization noise*.
- The more bits for quantization of a signal, the more closely the original signal is reproduced.

Quantization

- Using higher sampling frequency and more bits for quantization will produce better quality digital audio.
- But for the same length of audio, the file size will be much larger than the low quality signal.

Quantization

- The number of bits available to describe sampling values determines the resolution or accuracy of quantization.
- For example, if you have 8-bit analog to digital converters, the varying analog voltage must be quantized to 1 of 256 discrete values;
- a 16-bit converter has 65,536 values.

Nyquist Theorem

- A theorem which states that an analog signal waveform may be uniquely reconstructed, without error, from samples taken at equal time intervals.

Nyquist Theorem

- The sampling rate must be equal to, or greater than, twice the highest frequency component in the analog signal.

CSE 466 15

Nyquist Theorem

- Stated differently:
- The highest frequency which can be accurately represented is one-half of the sampling rate.

CSE 466 16

Error

- Sampling an analog signal can introduce ERROR.
- ERROR is the difference between a computed, estimated, or measured value and the true, specified, or theoretically correct value.

Nyquist Theorem

- By sampling at TWICE the highest frequency:
  - One number can describe the positive transition, and…
  - One number can describe the negative transition of a single cycle.

Nyquist Error-- aliasing

upper => sampling 6 times per cycle(fs=6f);
middle => sampling 3 times per cycle(fs=3f);
lower=> sampling 6 times in 5 cycles, from[1]

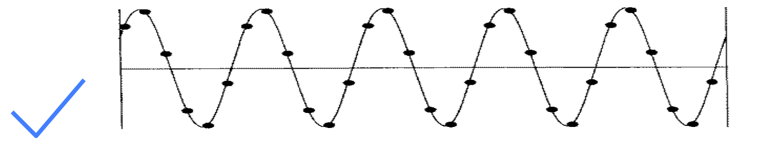


Figure 3.14 When a sine wave is sampled six times per cycle ($f_S = 6f$), the observed frequency is equal to the true frequency.

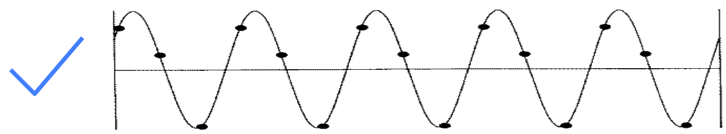


Figure 3.15 When a sine wave is sampled three times per cycle ($f_S = 3f$), the observed frequency is equal to the true frequency.

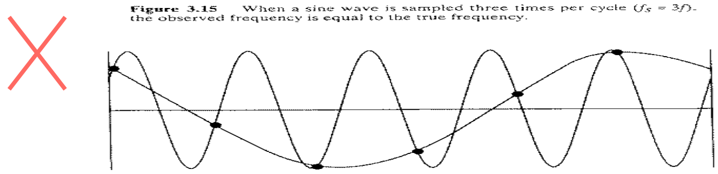


Figure 3.16 When a sine wave is sampled six times in five cycles, the observed sine wave has a much lower frequency than the original sine wave.

Digital Synthesis Overview

- Sound is created by manipulating numbers, converting those numbers to an electrical current, and amplifying result.
- Numerical manipulations are the same whether they are done with software or hardware.
- Same capabilities (components) as analog synthesis, plus significant new abilities

Digital Oscillators

- Everything is a Table
  - A table is an indexed list of elements (or values)
  - The index is the address used to find a value

Generate a Sine Tone Digitally (1)

- Compute the sine in real time, every time it is needed.
  - equation:

$$signal(t) = r\sin(\omega t)$$

  - t = a point in time; r = the radius, or amplitude of the signal; w (omega) = 2pi*f the frequency
  - Advantages: It's the perfect sine tone. Every value that you need will be the exact value from the unit circle.
  - Disadvantages: must generate every sample of every oscillator present in a synthesis patch from an algorithm. This is very expensive computationally, and most of the calculation is redundant.

Generate a Sine Tone Digitally (2)

- Compute the sine tone once, store it in a table, and have all oscillators look in the table for needed values.
  - Advantages: Much more efficient, hence faster, for the computer. You are not, literally, re-inventing the wheel every time.
  - Disadvantages: Table values are discrete points in time. Most times you will need a value that falls somewhere in between two already computed values.

Table Lookup Synthesis

- Sound waves are very repetitive.
- For an oscillator, compute and store one cycle (period) of a waveform.
- Read through the wavetable repeatedly to generate a periodic sound.

Changing Frequency

- The Sample Rate doesn't change within a synthesis algorithm.
- You can change the speed that the table is scanned by skipping samples.
- skip size is the increment, better known as the phase increment.

  phase increment is a very important concept

Algorithm for a Digital Oscillator

- Basic, two-step program:
  - $phase\_index = \mathrm{mod}_L(previous\_phase + increment)$
  - $output = amplitude \times wavetable[phase\_index]$
- $increment = \dfrac{TableLength \times DesiredFrequency}{SampleRate}$

If You're Wrong, it's Noise

- What happens when the phase increment doesn't land exactly at an index location in the table?
  - It simply looks at the last index location passed for a value.
    In other words, the phase increment is truncated to the integer.
- Quantization
- Noise
- The greater the error, the more the noise.

Interpolation

- Rather than truncate the phase location…
  - look at the values stored before and after the calculated phase location
  - calculate what the value would have been at the calculated phase location if it had been generated and stored.

  Interpolate
- More calculations, but a much cleaner signal.

Linear interpolation

- Interpolate between two audio samples

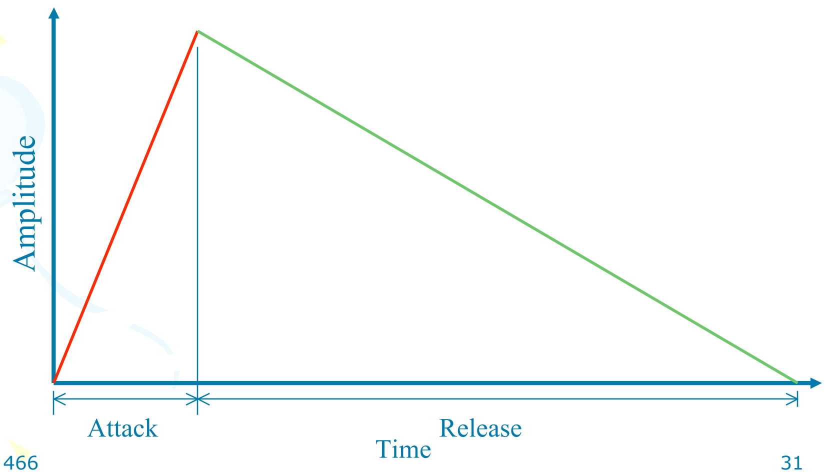
```
double inbetween = fmod(sample, 1);
return (1. – inbetween) * wave[int(sample)] +
        inbetween * wave[int(sample) + 1];
```

- More accurate, yet still efficient

Envelopes

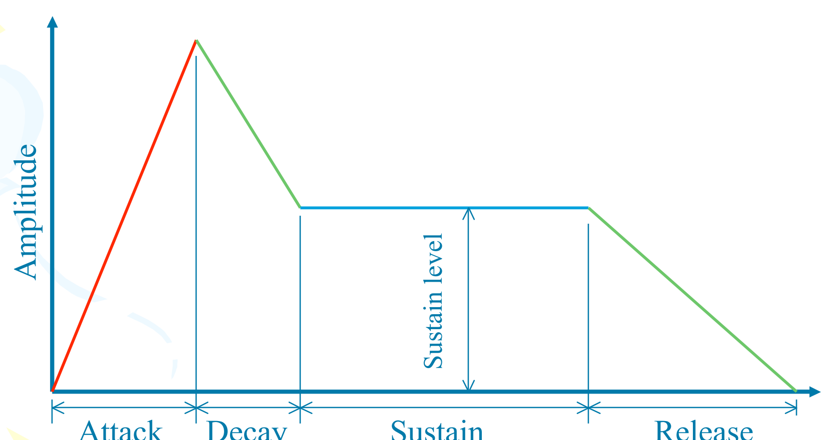
- We commonly will make samples with fixed amplitudes, then make a synthetic *envelope* for the sound event.

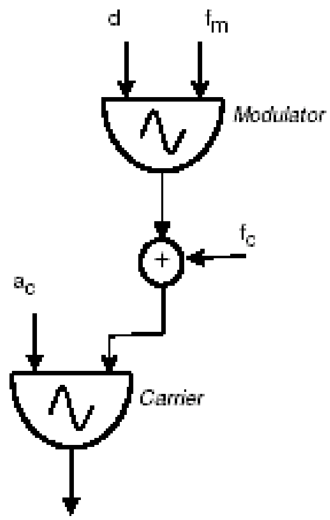
Attack and Release



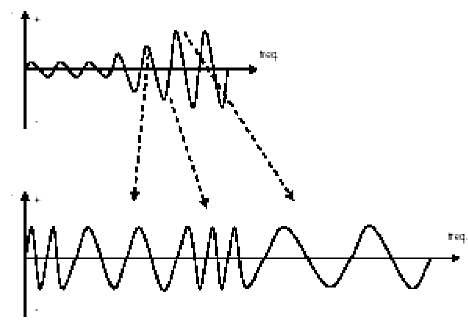
ADSR

- ADSR: Attack, decay, sustain, release

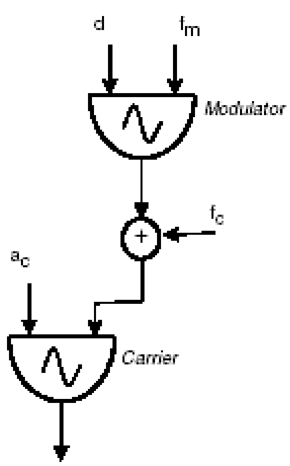




Frequency Modulation



FM: General Description



- Simple FM: carrier oscillator has its frequency modulated by the output of a modulating oscillator.
- Sidebands produced around carrier at multiples of modulating frequency.
  - Number generated depends on the amplitude of the modulator.

Modulator : Carrier Ratio

- Sidebands at C + and - (n * Modulator)
- Ratio of M:C determines whether spectrum is harmonic or not.
  - Simple integer ratio = harmonic
  - Non-integer ratio = inharmonic

Modulation Index and Bandwidth

- The *bandwidth* of the FM spectrum is the number of sidebands present.
- The bandwidth is determined by the *Modulation Index*
  - *I = depth of modulation / modulator*
  - *D* depth of modulation, which depends on the amount of amplitude applied to modulating oscillator. (*D = A* x *M*)
- If the index is above zero, then sidebands occur.

FM Voices

- Case 1: if f_c **is equal to any integer and** f_m **is equal to 1, 2, 3 or 4**, then the resulting timbre will have a distinctive pitch, because the offset carrier frequency will always be prominent.
- Case 2: if f_c **is equal to any integer and** f_m **is equal to any integer higher than 4**, then the modulation produces harmonic partials but the fundamental may not be prominent.
- Case 3: if f_c is equal to any integer and f_m is equal to 1, then the modulation produces a spectrum composed of harmonic partials; e.g. the ratio 1:1 produces a sawtooth-like wave.
- Case 4: if f_c **is equal to any integer and** f_m **is equal to any even number**, then the modulation produces a spectrum with some combination of odd harmonic partials; e.g. the ratio 2:1 produces a square-like wave.
- Case 5: if f_c **is equal to any integer and** f_m **is equal to 3**, then every third harmonic partial of the spectrum will be missing; e.g. the ratio 3:1 produces narrow pulse -like waves.
- Case 6: if f_c **is equal to any integer and** f_m **is not equal to an integer**, then the modulation produces non-harmonic partials; e.g. 2:1.29 produces a "metallic" bell sound

What we're going to do:

- Next week:
  - Talk to the sound driver
  - Play sound sample files
  - Build a table-driven oscillator
- In two weeks:
  - Implement game controller
  - Add emergent sound behavior