This pdf for ICM students only - ebook
and paperback available from amazon.com

# Introduction to Computer Music

## Week 10

**Instructor:** Prof. Roger B. Dannenberg

**Topics Discussed:** Pitch vs. Frequency, Loudness vs. Amplitude, Localization, Linearity, Reverberation, Echo, Equalization, Chorus, Panning, Dynamics Compression, Sample-Rate Conversion, Convolution Reverberation

This pdf for ICM students only - ebook
and paperback available from amazon.com

# Chapter 10

# Acoustics, Perception, Effects

**Topics Discussed:** Pitch vs. Frequency, Loudness vs. Amplitude, Localization, Linearity, Reverberation, Echo, Equalization, Chorus, Panning, Dynamics Compression, Sample-Rate Conversion, Convolution Reverberation

## 10.1 Introduction

Acoustics and perception are very different subjects: acoustics being about the physics of sound and perception being about our sense and cognitive processing of sound. Though these are very different, both are very important for computer sound and music generation. In sound generation, we generally strive to create sounds that are similar to those in the real world, sometimes using models of how sound is produced by acoustic instruments. Alternatively, we may try to create sound that will cause a certain aural impression. Sometimes, an understanding of physics and perception can inspire sounds that have no basis in reality.

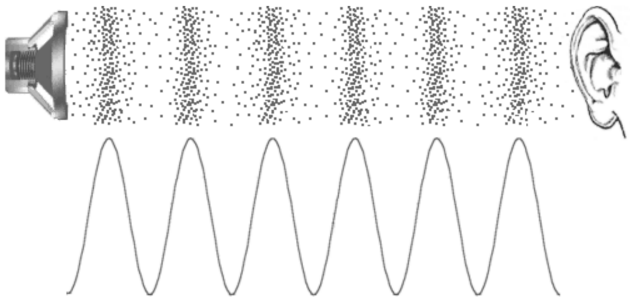

Figure 10.1: Sound is vibration or air pressure fluctuations.
(Credit: mediacollege.com)

Sound is vibration or air pressure fluctuations (see Figure 10.1). We can hear small pressure variations, e.g. 0.001 psi ($lbs/in^2$) for loud sound. (We are purposefully using English units of pounds and inches because if you inflate a bicycle tire or check tires on an automobile – at least in the U.S. – you might have some idea of what that means.) One psi $\approx$ 6895 Pascal (Pa), so 0.001 psi is about 7 Pascal. At sea level, air pressure is 14.7 pounds per square inch, while the cabin pressure in an airplane is about 11.5 psi, so 0.001 (and remember that is a *loud* sound) is a tiny tiny fraction of the nominal constant pressure around us. Changes in air pressure deflect our ear drum. The amplitude of deflection of ear drum is

This pdf for ICM students only - ebook
and paperback available from amazon.com

about diameter of hydrogen atom for the softest sounds. So we indeed have a extremely sensitive ears!

What can we hear? The frequencies that we hear range over three orders of magnitude from about 20 to 20 kHz. As we grow older and we are exposed to loud sounds, our high frequency hearing is impaired, and so the actual range is typically less.

Our range of hearing, in terms of loudness, is about 120 dB, a power ratio of $10^{12}$, measured from threshold of hearing to threshold of pain (discomfort from loud sounds). In practical terms, our dynamic range is actually limited and often determined by background noise. Listen to your surroundings right now and listen to what you can hear. Anything you hear is likely to *mask* even softer sounds, limiting your ability to hear them and reducing the effective dynamic range of sounds you can hear.

We are very sensitive to the amplitude spectrum. Figure 10.2 shows a spectral view that covers our range of frequency, range of amplitude, and suggests that the shape of the spectrum is something to which we are sensitive.

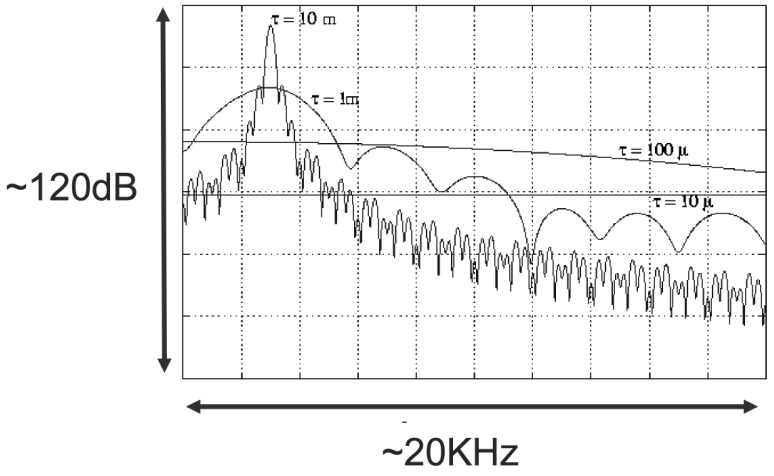


Figure 10.2: The actual spectrum shown here is not important, but the graph covers the range of audible frequencies (about 20KHz) and our range of amplitudes (about 120 dB). We are very sensitive to the shape of the spectrum.

Real-world sounds are richly complicated and full of information and detail. We have seen many synthesis algorithms that produce very clean, simple, specific spectra and waveforms. These *can* be musically interesting, but they are not characteristic of sounds in the "real world." This is important; *if you want to synthesize musical sounds that are pleasing to the ear*, it is important to know that, for example, real-world sounds are not simple sinusoids, they do not have constant amplitude, and so on.

Let's consider some "real-world" sounds. First, noisy sounds, such as the sound of "shhhh," tend to be broadband, meaning they contain energy at almost all frequencies, and the amount of energy in any particular frequency, or the amplitude at any particular time, is random. The overall spectrum of noisy sounds looks like the top spectrum in Figure 10.3.

Percussion sounds on the other hand, such as a thump, bell clang, ping, or knock, tend to have resonances, so the energy is more concentrated around certain frequencies. The middle picture of Figure 10.3 gives the general spectral appear-

This pdf for ICM students only - ebook
and paperback available from amazon.com

ance of such sounds. Here, you see resonances at different frequencies. Each resonance produces an exponentially decaying sinusoid. The decay causes the spectral peak to be wider than that of a steady sinusoid. In general, the faster the decay, the wider the peak. It should also be noted that, depending on the material, there can be non-linear coupling between modes. In that case, the simple model of independent sinusoids with exponential decay is not exact or complete (but even then, this can be a good approximation.)

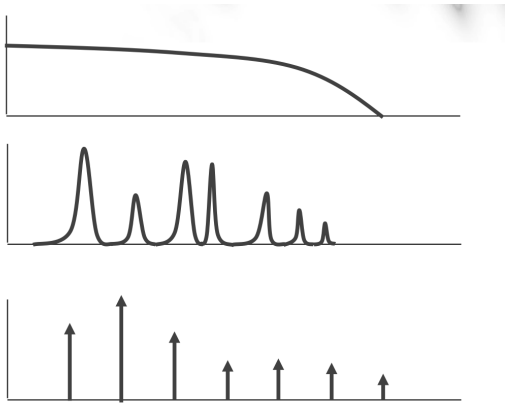

Figure 10.3: Some characteristic spectra. Can you identify them? Horizontal axis is frequency. Vertical axis is magnitude (amplitude). The top represents noise, with energy at all frequencies. The middle spectrum represents a percussion sound with resonant or "characteristic" frequencies. This might be a bell sound. The bottom spectrum represents a musical tone, which has many harmonics that are multiples of the first harmonic, or *fundamental*.

Figure 10.4 shows some modes of vibration in a guitar body. This is the top of an acoustic guitar, and each picture illustrates how the guitar top plate flexes in each mode. At the higher frequencies (e.g. "j" in Figure 10.4), you can see patches of the guitar plate move up while neighboring patches move down. If you tap a guitar body, you will "excite" these different modes and get a collection of decaying sinusoids. The spectrum might look something like the middle of Figure 10.3. (When you play a guitar, of course, you are strumming strings that have a different set of modes of vibration that give sharper peaks and a clearer sense of pitch.)

Pitched sounds are often called *tones* and tend to have harmonically related sinusoids. For example, an "ideal" string has characteristic frequencies that form a harmonic series, as illustrated in Figure 10.5. This can be seen as a special case of vibrating objects in general, such as the guitar body (Figure 10.4.) The vibrating string just happens to have harmonically related mode frequencies.

We know from previous discussions that if the signal is purely periodic, then it has harmonically related sinusoids. In other words, any periodic signal can be decomposed into a sum of sinusoids that are all multiples of some fundamental frequency. *In physical systems, periodicity is characteristic of some kind of mechanical oscillator that is driven by an outside energy source.* If you bow a string, sing a tone, or blow into a clarinet, you drive an oscillator with a constant source of energy, and almost invariably you end up with a stable periodic oscillation.

In summary, there are two ways to obtain harmonic or near-harmonic partials that are found in musical tones: One is to strike or pluck an object that has modes of vibration that just happen to resonate at harmonic frequencies. An example is a piano string or chime. The other is to drive oscillation with a steady input of

This pdf for ICM students only - ebook
and paperback available from amazon.com

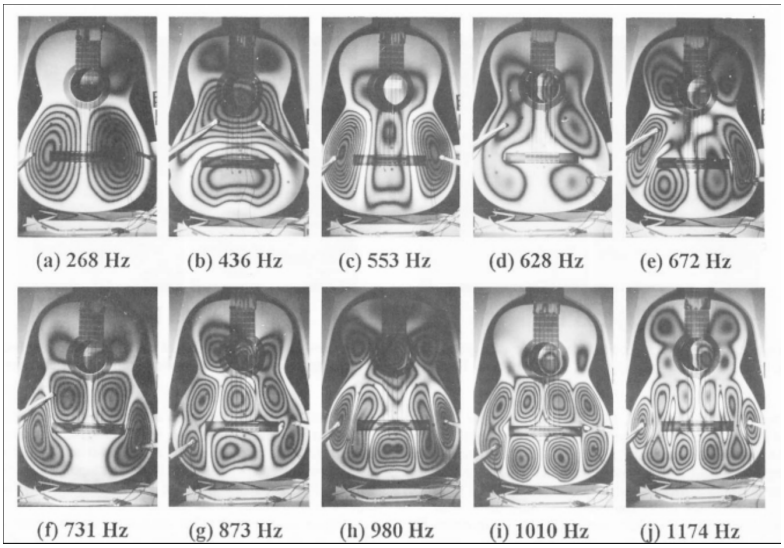


Figure 10.4: Modes of vibration in an acoustic guitar body.
(Credit: http://michaelmesser.proboards.com/thread/7581/resonator-cone-physics)

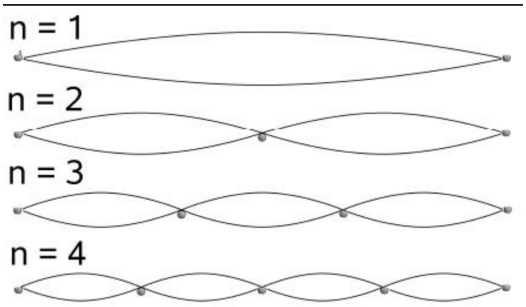


Figure 10.5: Modes of vibration in a stretched string. In an "ideal" string, the characteristic frequencies are multiples of the frequency of the first mode. The modes of vibration are independent, so the "sound" of the string is formed by the superposition of all the vibrating modes, resulting in a harmonic spectrum.
(Credit: phycomp.technion.ac.il)

This pdf for ICM students only - ebook
and paperback available from amazon.com

energy, as in a bowed string or wind instrument, giving rise to a periodic signal. In addition to these sounds called "tones," we have inharmonic spectra and noisy spectra.

## 10.2 Perception: Pitch, Loudness, Localization

Having focused mainly on physical properties of sound and sound sources, we now our attention to our *perception* of sound. It is important to keep in mind that our perception does not directly correspond to the underlying physical properties of sound. For example, two sounds with the same perceived loudness might have very different amplitudes, and two pitch intervals perceived to be the same musically could be not at all the same in terms of frequency differences.

### 10.2.1 Pitch Perception

Pitch is fundamental to most music. Where does it come from? How sensitive are ears to pitch? Our sense of pitch is strongly related to frequency. Higher frequencies mean higher pitch. Our sense of pitch is enhanced by harmonic partials. In fact, the connection is so strong that we are unable to hear individual partials. In most cases, we collect all of these partials into a single tone that we perceive as a single pitch, that of the lowest partial.

Pitch perception is approximately logarithmic, meaning that when pitch doubles, you hear the pitch interval of one octave. When it doubles again, you hear the same interval even though now the frequency is four times as high. If we divide the octave (factor of 2 in frequency) into 12 log-spaced intervals, we get what are called musical semitones or half-steps. This arrangement of 12 equal ratios per octave (each one $\sqrt[12]{2}$) is the basis for most Western music, as shown in the keyboard (Figure 10.6).

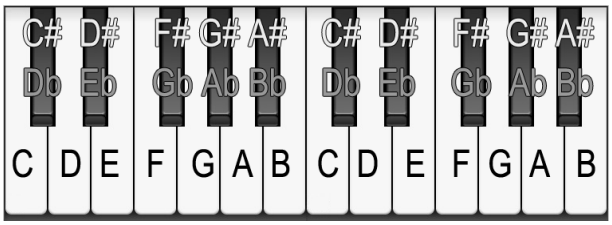


Figure 10.6: A piano keyboard. The ratio between frequencies of adjacent keys is $\sqrt[12]{2}$.

(Credit: http://alijamieson.co.uk/2017/12/03/describing-relationship-two-notes/)

We can divide a semitone ( $\sqrt[12]{2}$) into 100 log-spaced frequency intervals called cents. Often cents are used in studies of tuning and intonation. We are sensitive to about 5 cents, which is a ratio of about 0.3%. No wonder it is so hard for musicians to play in tune!

### 10.2.2 Amplitude

The term *pitch* refers to *perception* while *frequency* refers to *objective* or *physical* repetition rates. In a similar way, *loudness* is the perception of *intensity* or *amplitude* of sound. Here, we introduce some terminology and units of measurement for amplitude. Below, we will return to the topic of loudness.

This pdf for ICM students only - ebook
and paperback available from amazon.com

There are multiple ways of defining amplitude. One definition is: "a measure of a periodic function's change over a single period." Figure 10.7 illustrates the three different ways amplitude may be measured:

1. The peak amplitude
2. The peak-to-peak amplitude
3. The root mean square (RMS) amplitude

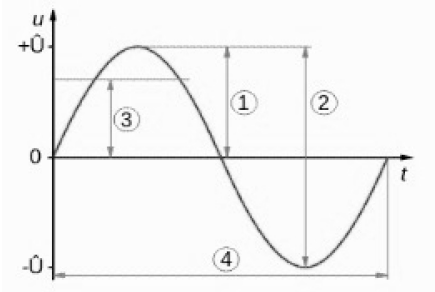


Figure 10.7: Three different ways of measuring amplitude. Note: the 4 on the Figure indicates the length of one period.
Credit: Wikipedia

In specifying amplitude in code, we are mostly using the peak amplitude of a signal. Thus, the amplitude of a sine wave, $y = \sin(t)$, is 1, and the amplitude of $y = 2\cos(t)$ is 2. When we multiply an oscillator in Nyquist by an envelope, we're varying the amplitude of that oscillator over time.

### Volume

There is no one formal definition for what the volume of sound means. Generally, volume is used as a synonym for loudness. In music production, adjusting the volume of a sound usually means to move a volume fader on e.g. a mixer. Faders can be either linear or logarithmic, so again, it is not exactly clear what volume means (i.e. is it the fader position, or the perceived loudness?).

### Power

Power is the amount of work done per unit of time; e.g. how much energy is transferred by an audio signal. Hence, the average power $\bar{P}$ is the ratio of energy E to time t: $\bar{P} = E/t$. Power is usually measured in watts (W); which are joules (J) per second: $J/s = W$. The power of a sound signal is proportional to the squared amplitude of that signal. Note that in terms of relative power or amplitude, it does not really matter whether we think of amplitude as the instantaneous amplitude (i.e. the amplitude at a specific point in time), or the peak, the peak-to-peak, or the root mean square amplitude over a single period. It makes no significant difference to ratios.

### Pressure

In general, the pressure $p$ is the amount of force $F$ applied perpendicularly (normal) to a surface area divided by the size $a$ of that area: $p = F/a$. Pressure is normally measured in pascal (Pa), which is newtons (N) per square meter. Thus, $Pa = N/m^2$.

This pdf for ICM students only - ebook
and paperback available from amazon.com

### Intensity

Intensity $I$ is the energy $E$ flowing across a unit surface area per unit of time t: $I = \frac{E}{a}\frac{1}{t}$. For instance, the energy from a sound wave flowing to an ear. As the average power $\bar{P} = E/t$, we can express the intensity as $I = \bar{P}/a$, meaning the power flowing across a surface of area $a$. The standard unit area is one square meter, and therefore we measure intensity in $W/m^2$; i.e. watts per square meter.

### Range of Human Hearing

Just as the frequency range of the human ear is limited, so is the range of intensity. For a very sensitive listener, and a 1 kHz frequency:

- The threshold of hearing $t_h = 10^{-12} W/m^2$
- The limit of hearing (due to pain and injury caused by sound) is $l_h = 1 W/m^2$ (the threshold of pain).

The perceptual range of intensity for a 1 kHz frequency is an impressive $l_h/t_h = 1/10^{-12} = 10^{12}$, or a trillion to one.

### The Bel Scale

The bel, abbreviated B (note capitalizations), is a sound intensity scale named in honor of the renowned Alexander Graham Bell (1847-1922). Given the enormous range of human hearing, one bel is defined as a factor of 10 in intensity or power as follows:

$$bels = \log_{10} \frac{I}{I_0}$$

where $I$ is intensity or power and $I_0$ is a reference. In other words, bel is the log ratio of intensity or power. Measuring the range of intensity for human hearing $H$ in bels, we get

$$H \text{ bel} = \log_{10} \frac{l_h}{t_h} = 12$$

For sound, we measure intensity, but in other situations, such as an electrical signal, we measure total *power*, e.g. as watts, since there is no area to divide by. Since bel (B) is a ratio, the units cancel out, so we can use the bel for both intensity and power.

### The Decibel Scale (dB)

Using just a range $H = 12$ to express the entire range of hearing is inconvenient, so it is customary to use decibels, abbreviated dB, instead. One bel is 10 dB. The intensity range of human hearing is therefore 120 dB.

### Decibels for Comparing Sound Intensity Levels

Decibels are most commonly used to compare the intensity levels of two sounds. As shown in the definition of *bels* above, we take the intensity $I$ of some sound and compare it to a reference intensity $I_0$. This reference is arbitrary. For example, instead of using $t_h$ as the reference intensity, we could use the limit of hearing $l_h$, and thus measure down from pain instead of measuring up from silence. With this reference, the threshold of hearing becomes -120 dB.

This pdf for ICM students only - ebook
and paperback available from amazon.com

### Decibels for Comparing Amplitude Levels

A common confusion arises when working with intensity and amplitudes. We just saw that given an intensity ratio $r$, we can express the ratio in decibels using $10\log_{10} r$. This also works for a ratio of power because power and intensity are proportional. However, when we are working with amplitudes, this formula does not apply. Why? Because the decibel is a measure of power or intensity ratio. Since power is proportional to the square of amplitude, a different formula must be used for amplitudes. For two amplitudes $A$ and $B$, we can use the power formula if we square them. Then we can simplify the expression:

$$dB = 10\log_{10}\frac{A^2}{B^2} = 10\log_{10}(\frac{A}{B})^2 = 20\log_{10}\frac{A}{B}$$

**Main idea**: for power or intensity we use $10\log_{10} ratio$, but for amplitudes, we must use $20\log_{10} ratio$.

### Other dB Variants

You see the abbreviation dB in many contexts.

- **dBa** uses the so-called A-weighting to account for the relative loudness perceived by the human ear; based on equal-loudness contours.
- **dBb** and **dBc** are similar to **dBa**, but apply different weighting schemes.
- **dBm** (or $dBmW$) is the power ratio in dB of a measured power referenced to one milliwatt (mW); used e.g. in radio, microwave and fiber optic networks as a measure of absolute power.
- **dB SPL** – see below.

### Amplitude and Gain in Recording Equipment

In a music studio, we usually want to measure down from the limit of the loudest sound that can be recorded without introducing distortion from the recording equipment. This is why it is standard for e.g. mixing consoles and music software to use volume faders and meters that show negative dB. In this context, 0 dB is the upper limit for recording without distortion; let's call it the limit of recording = $l_r$. Hence, the loudest sound we may record without any distortion has intensity $I = l_r$, and the corresponding dB is $10\log_{10}\frac{l_r}{l_r} = 0$ dB. Note that it is customary for producers of recording gear to leave some amount of head room at the top of this scale to safeguard against distortion, which is why you'll see some positive dB values above 0 dB on meters and faders.

### Sound Pressure

Measuring the intensity of a sound signal is usually not practical (or possible); i.e. measuring the energy flow over an area is tricky. Fortunately, we can measure the average variation in pressure. Pressure is the force applied normal to a surface area, so if we sample a sufficiently large area we'll get a decent approximation; this is exactly what a microphone does!

It is worth noting that we can relate sound pressure to intensity by the following ratio,

$$\frac{\Delta p^2}{V\delta}$$

This pdf for ICM students only - ebook
and paperback available from amazon.com

where $\Delta p$ is the variation in pressure, $V$ is the velocity of sound in air, and $\delta$ is the density of air. What this tells us is that intensity is proportional to the square of the variation in pressure.

#### dB SPL (Sound Pressure Level)

Sound pressure level is defined as the average pressure variation per unit area. The dB SPL is defined as $20\log_{10}\frac{P}{P_0}$, where the reference pressure, $P_0$, is 0.00005 Pa, which is approximately the threshold of hearing at 1 kHz.

#### The Microphone: Measuring Sound Pressure

Most microphones use electromagnetic induction to transform the sound pressure applied to a diaphragm into an electrical signal; as shown in Figure 10.8.

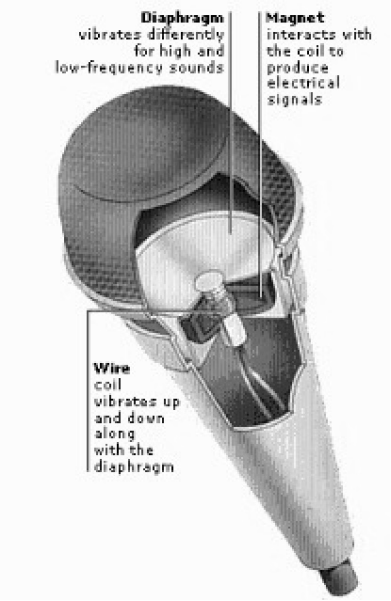


Figure 10.8: Microphone.
(Credit: infoplease.com)

For more information on loudness, dB, intensity, pressure etc., see *Musimathics* Vol. 1 [**?**].

### 10.2.3 Loudness

Loudness is a perceptual concept. Equal loudness does necessarily result from equal amplitude because the human ear's sensitivity to sound varies with frequency. This sensitivity, the loudness, is depicted in equal-loudness contours for the human ear; often referred to as the Fletcher-Munson curves. Fletcher and Munson's data were revised to create an ISO standard. Both are illustrated in Figure 10.9 below. Each curve traces changes in amplitude required to maintain equal loudness as the frequency of a sine tone varies. In other words, each curve depicts amplitude as a function of frequency at a constant loudness level.

Loudness is commonly measured in phons.

This pdf for ICM students only - ebook
and paperback available from amazon.com

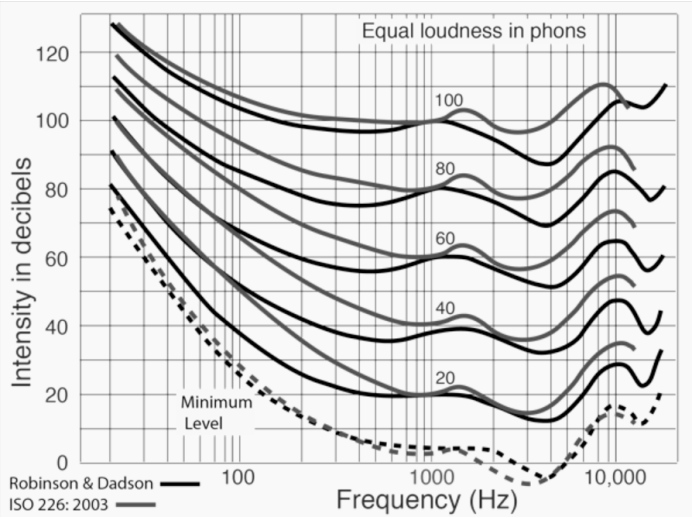


Figure 10.9: Equal-loudness contours (left-most / lighter) from ISO 226:2003 revision. So-called Fletcher-Munson curves, as measured by Robinson and Dadson, shown (right-most / darker) for comparison.
Credit: http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/eqloud.html

### Phon

Phon expresses the loudness of a sound in terms of a reference loudness. That is, the phon level of a sound, A, is the dB SPL (defined earlier) of a reference sound—of frequency 1 kHz—that has the same (perceived) loudness as A. Zero phon is the limit of audibility of the human ear; inaudible sounds have negative phon levels.

### Rules of Thumb

*Loudness* is mainly dependent on amplitude, and this relationship is approximately logarithmic, which means equal ratios of amplitude are more-or-less perceived as equal increments of loudness. We are very sensitive to small ratios of frequency, but we are not very sensitive to small ratios of amplitude. To double loudness you need about a 10-fold increase in intensity or about 10 dB. We are sensitive to about 1 dB of amplitude ratio with careful listening. 1 dB is a change of about 12%.

The fact that we are not so sensitive to amplitude changes is important to keep in mind when adjusting amplitude. If you synthesize a sound and it is too soft, try scaling it by a factor of 2 (about 6 dB). People are often tempted to make small changes, e.g. multiply by 1.1 to make something louder, but in most cases, a 10% change is not even audible (less than 1 dB).

Loudness also depends on frequency because we are not so sensitive to very high and very low frequencies. The Fletcher-Munson Curve contours shown above are lowest around 4 kHz where we are most sensitive. The curve is low here because a low amplitude at 4 kHz sounds as loud as higher amplitudes at other frequencies. As we move to even higher frequencies over on the right, we become less sensitive once again. The curves trace the combinations of frequency and amplitude that sound equally loud. If you sweep a sinusoid from low frequency to high frequency at equal amplitude, you will hear that the sound appears to get louder until you hit around 4 kHz, and then the sound begins to get quieter.

This pdf for ICM students only - ebook
and paperback available from amazon.com

Depending on how loud it is, you might stop hearing it at some point before you hit 20 kHz, even if you can hear loud sounds at 20 kHz.

The red (or gray) lines in Figure 10.9 show an update to the Fletcher-Munson Curve based on new measurements of typical human hearing. It should be noted that these curves are based on sinusoids. In most music, low pitches actually consist of a mixture of harmonics, some of which can have rather high frequencies. So even if you cannot hear the fundamental because it is too low or too quiet, you might hear the pitch, which is implied by the upper partials (to which you are more sensitive). This is also why, on your laptop, you can hear a piano tone at the pitch $C_2$ (below the bass clef, with a fundamental frequency of about 65 Hz), even though your laptop speaker is barely able to produce any output at 65 Hz. You might try this SAL command, which plays a piano tone followed by a sine tone:

```
play seq(note(pitch: c2), osc(c2))
```

It is instructive to listen to this *using a quiet volume setting* with good headphones. You should hear the two tones at the same pitch. Then, listen to the same sounds on your built-in laptop speaker. You will probably not hear the second tone, indicating that your computer cannot produce frequencies that low. And yet, the pitch of the first tone, even with *no fundamental frequency present* retains the same pitch!

### 10.2.4 Localization

Localization is the ability to perceive direction of the source of a sound and perhaps the distance of that sound. We have multiple cues for localization, including relative amplitude, phase or timing, and spectral effects of the pinnae (outer ears). Starting with amplitude, if we hear something louder in our right ear than our left ear, then we will perceive that the sound must be coming from the right, all other things be equal.

We also use relative phase or timing for localization: if we hear something arrive earlier in our right ear than our left ear, then we will perceive that sound as coming from the right.

The third cue is produced by our pinnae or outer ears. Sound reflects off the pinnae, which are very irregularly shaped. These reflections cause some cancellation or reinforcement at particular wavelengths, depending on the direction of the sound source and the orientation of our head. Even though we do not know the exact spectrum of the source sound, our amazing brain is able to disentangle all this information and compute something about localization. This is especially important for the perception of elevation, i.e. is the sound coming from ahead or above? In either case, the distance to our ears is the same, so whether the source is ahead or above, there should be no difference in amplitude or timing. The only difference is in spectral changes due to our outer ears and reflections from our shoulders.

All of these effects or cues can be described in terms of filters. Taken together, these effects are sometimes called the HRTF, or *Head-Related Transfer Function*. (A "transfer function" describes the change in the spectrum from source to destination.) You might have seen some artificial localization systems, including video games and virtual reality application systems based on HRTF. The idea is, for each source to be placed in a virtual 3D space, compute an appropriate HRTF and apply that to the source sound. There is a different HRTF for the left ear and right ear, and typically the resulting stereo signal is presented through headphones. Ideally, the headphones are tracked so that the HRTFs can be recomputed as the head turns and the angles to virtual sources change.

This pdf for ICM students only - ebook
and paperback available from amazon.com

Environmental cues are also important for localization. If sounds reflect off walls, then you get a sense of being in a closed space, how far away the walls are, and what the walls are made of. Reverberation and ratio of reverberation to direct sound are important for distance estimation, especially for computer music if you want to create the effect of a sound source fading off into the distance. Instead of just turning down the amplitude of the sound, the direct or dry sound should diminish faster than the reverberation sound to give the impression of greater distance.

Finally, knowledge of the sound source, including vision, recognition etc. is very important to localization. For example, merely placing a silent loudspeaker in front of a listener wearing headphones can cause experimental subjects to localize sound at the loudspeaker, ignoring whatever cues are present in the actual sound!

### 10.2.5 More Acoustics

The speed of sound is about 1 ft/ms. Sound travels at different speeds at different altitudes, different temperatures and different humidities, so it is probably more useful to have a rough idea of the speed of sound than to memorize a precise number. (But for the record, the speed of sound is 343 m/s in dry air at 20° C.) Compared to the speed of light, sound is painfully slow. For example, if you stand in front of a wall and clap, you can hear that the sound reflects from the wall surfaces, and you can perceive the time for the clap to travel to the wall and reflect back. Our auditory system merges multiple reflections when they are close in time (usually up to about 40 ms), so you do not perceive echoes until you stand back 20 feet or so from the reflecting wall.

In most listening environments, we do not get just one reflection, called an echo. Instead, we get a diffuse superposition of millions of echos as sound scatters and bounces between many surfaces in a room. We call this *reverberation*. In addition to reflection, sound refracts (bends around objects). Wavelengths vary from 50 feet to a fraction of an inch, and diffraction is more pronounced at lower frequencies, allowing sound to bend around objects.

Linearity is a very important concept for understanding acoustics. Let's think about the transmission of sounds from the source of sound to the listener. We can think of the whole process as a function $F$ shown in the equations below, where $y = F(x)$ means that source $x$ is transformed by the room into $y$ at the ear of the listener. *Linearity* means that if we increase the sound at the source, we will get a proportional increase at the listening side of channel $F$. Thus, $F(ax) = aF(x)$. The other property, sometimes called the *superposition*, is that if we have two sources: pressure signals $x_1$ and $x_2$, and play them at the same time, then the effect on the listener will be the sum of individual effect from $x_1$ and $x_2$:

$$F(ax) = aF(x), \quad F(x_1 + x_2) = F(x_1) + F(x_2)$$

Why does linearity matter? First, air, rooms, performance spaces are very linear. Also, many of processes we used on sounds such as filters are designed to be linear. Linearity means that if there are two sound sources playing at the same time, then the signal at the listening end is equivalent to what you get from one sound plus what you get from the other sound. Another interesting thing about linearity is that we can decompose a sound into sinusoids or components (i.e. compute the Fourier transform). If we know what a linear system does to each one of those component frequencies, then by the superposition principle, we know what the system does to any sound, because we can: break it up into component frequencies, compute the transfer function at each of the frequencies

This pdf for ICM students only - ebook
and paperback available from amazon.com

and sum the results together. That is one way of looking at what a filter does. A filter associates different scale factors with each frequency, and because filters are linear, they weight or delay frequencies differently but independently. If we add two sounds and put them through the filter, the result is equivalent to putting the two sounds through the filter independently and summing the results.

### 10.2.6 Summary: Acoustics and Perception

Now we summarize this discussion of acoustics and perception. Acoustics refers to the physics of sound, while perception concerns our sense of sound. As summarized in Figure 10.10, pitch is the perception of (mainly) frequency, and loudness is the perception (mainly) of amplitude or intensity. Our perception of pitch and loudness is roughly logarithmic with frequency and amplitude. This relationship is not exact, and we saw in the Fletcher-Munson Curve that we are more sensitive to some frequencies than others. Generally, everything else we perceive is referred to as timbre, and you can think of timbre as (mainly) the perception of spectral shape. In addition to these properties, we can localize sound in space using various cues that give us a sense of direction and distance.

| Perception | Acoustic Phenomenon |
|---|---|
| Pitch | Frequency (20-20 kHz range) |
| Loudness | Intensity (120 dB range) |
| Timbre | Spectrum (and other) |

Figure 10.10: A comparison of concepts and terms from perception (left column) to acoustics (right column).

Struck objects typically exhibit characteristic frequencies with exponential decay rates (each mode of vibration has its own frequency and decay). In contrast, driven oscillators typically exhibit almost exactly periodic signals and hence harmonic spectra.

Our discussion also covered the speed of sound (roughly 1 ft/ms), transmission of sound as equivalent to filtering and the superposition principle.

## 10.3 Effects and Reverberation in Nyquist

There are a lot of effects and processes that you can apply to sound with Nyquist. Some are described here, and you can find more in the *Nyquist Reference Manual*. There are also many sound processing functions you can install with the Nyquist Extension Manager (in the NyquistIDE).

### 10.3.1 Delay or Echo

In Nyquist, we do not need any special unit generator to implement delay. We can directly create delay simply by adding sounds using an offset. Recall that Nyquist `sounds` have a built-in starting time and duration which are both immutable, so applying a shift operator to a sound does not do anything. However, the `cue` behavior takes a sound as parameter and returns a new sound that has been shifted according to the environment. So we usually combine `cue` with the shift operator `@`, and a delay expression has the form:

`cue(`*sound*`) @` *delay*

This pdf for ICM students only - ebook
and paperback available from amazon.com

### 10.3.2 Feedback Delay

An interesting effect is to not only produce an echo, but to add an attenuated copy of the output back into the input of the effect, producing a series of echoes that die away exponentially. There is a special unit generator in Nyquist called `feedback-delay` with three parameters: `feedback-delay(`*sound*`,` *delay*`,` *feedback*`)`. Figure 10.11 shows the echo algorithm: The input comes in and is stored in memory in a first-in-first-out (FIFO) queue; samples at the end of the buffer are recycled by adding them to the incoming sound. This is an efficient way to produce many copies of the sound that fade away. The *delay* parameter must be a number (in seconds). It is rounded to the nearest sample to determine the length of the delay buffer. The amount of *feedback* should be less than one to avoid an exponential increase in amplitude.

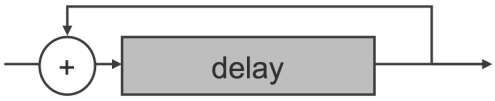


Figure 10.11: Echo algorithm in Nyquist.

Note that the duration of the output of this unit generator is equal to the duration of the input, so if the input is supposed to come to an end and then be followed by multiple echos, we need to append silence to the input source to avoid a sudden ending. The example below uses `s-rest()` to construct 10 seconds of silence, which follows the *sound*.

```
feedback-delay(seq(sound, s-rest(10)), delay, feedback)
```

In principle, the exponential decay of the `feedback-delay` effect never ends, so it might be prudent to use an envelope to smoothly bring the end of the signal to zero:

```
feedback-delay(seq(sound, s-rest(10)), delay, feedback)  *
pwlv(1,  d + 9, 1,  d + 10, 0)
```

, where *d* is the duration of *sound*. The sound is extended with 10 seconds of silence, so the envelope remains at 1 for the sound duration plus 9 seconds, then linearly falls to zero at sound duration + 10.

### 10.3.3 Comb Filter

Consider a feedback delay with a 10 ms delay. If the input is a sinusoid with 10 ms period (100 Hz), the echoes superimpose on one another and boost the amplitude of the input. The same happens with 200 Hz, 300 Hz, 400 Hz, etc. sinusoids because they all repeat after 10 ms. Other frequencies will produce echoes that do not add constructively and are not boosted much. Thus, this feedback delay will act like a filter with resonances at multiples of a fundamental frequency, which is the reciprocal of the delay time. The frequency response of a comb filter looks like Figure 10.12. Longer decay times gives the comb filter sharper peaks, which means the output has a more definite pitch and longer "ring."

The code below shows how to apply a comb filter to *sound* in Nyquist. A comb filter emphasizes (resonates at) frequencies that are multiples of the *hz* parameter. The decay time of the resonance is given by *decay*. The *decay* may be a sound or a number. In either case, it must also be positive. The resulting sound will

This pdf for ICM students only - ebook
and paperback available from amazon.com

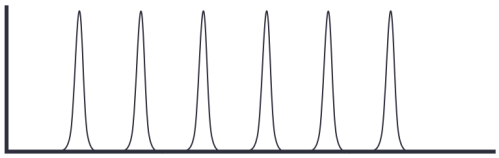

Figure 10.12: Filter response of a comb filter. The horizontal axis is frequency and the vertical axis is amplitude. The comb filter has resonances at multiples of some fundamental frequency.

have the start time, sample rate, etc. of *sound*. One limitation of `comb` is that the actual delay will be the closest integer number of sample periods to 1/*hz*, so the resonance frequency spacing will one that divides the sample rate evenly.

`comb(`*sound*`, `*decay*`, `*hz*`)`

### 10.3.4 Equalization

Equalization is generally used to adjust spectral balance. For example, we might want to boost the bass, or boost the high frequencies, or cut some objectionable frequencies in the middle range. The function `nband(`*input, gains*`)` takes an array of gains, one for each band, where the bands are evenly divided across the 20–20kHz range. An interesting possibility is using computed control functions to make the equalization change over time.

The Equalizer Editor in Nyquist provides a graphical equalizer interface for creating and adjusting equalizers. It has a number of sliders for different frequency bands, and you can slide them up and down and see graphically what the frequency response looks like. You can use this interface to create functions to be used in your code. Equalizers are named `eq-0`, `eq-1`, etc., and you select the equalizer to edit using a pull-down menu. The "Set" button should be use to record changes.

The following expression in Nyquist is a fixed- or variable-parameter, second-order midrange equalization (EQ) filter:

`eq-band(`*signal*`, `*hz*`, `*gain*`, `*width*`)`

The *hz* parameter is the center frequency, *gain* is the boost (or cut) in dB, and width is the half-gain width in octaves. Alternatively, *hz*, *gain*, and *width* may be sounds, but they must all have the same sample rate, e.g. they should all run at the control rate or at the sample rate.

You can look up filters in the Nyquist manual for many more filters and options.

### 10.3.5 Chorus

The chorus effect is a very useful way to enrich a simple and dry sound. Essentially, the "chorus" effect is a very short, time-varying delay that is added to the original sound. Its implementation is shown in Figure 10.13. The original sound passes through the top line, while a copy of the sound with some attenuation is added to the sound after a varying delay, which is indicated by the diagonal arrow.

To implement chorus in Nyquist, you need to first load library `time-delay-fns`[1] and then call the `chorus` function as shown below.

[1] In the future, `time-delay-fns` will be an extension installed with the Extension Manager.

This pdf for ICM students only - ebook
and paperback available from amazon.com

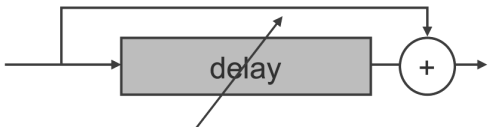


Figure 10.13: Chorus effect algorithm. A signal is mixed with a delayed copy of itself. The delay varies, typically by a small amount and rather slowly.

```
chorus(sound, delay: delay, depth: depth, rate: rate,
       saturation: saturation, phase: phase)
```

Here, a chorus effect is applied to *sound*. All parameters may be arrays as usual. The chorus is implemented as a variable delay modulated by a sinusoid shifted by *phase* degrees oscillating at *rate* Hz. The sinusoid is scaled by *depth*. The delayed signal is mixed with the original, and *saturation* gives the fraction of the delayed signal (from 0 to 1) in the mix. Default values are *delay* = 0.03, *depth* = 0.003, `rate` = 0.3, *saturation* = 1.0, and *phase* = 0.0 (degrees).

See also the *Nyquist Reference Manual* for the functions `stereo-chorus` and `stkchorus`.

### 10.3.6 Panning

Panning refers to the simulation of location by splitting a signal between left and right (and sometimes more) speakers. When panning a mono source from left to right in a stereo output, you are basically adjusting the volume of that source in the left and right channels. Simple enough. However, there are multiple reasonable ways of making those adjustments. In the following, we shall cover the three most common ones.

A typical two-speaker stereo setup is depicted in Figure 10.14; the speakers are placed symmetrically at $\pm 45$-degree angles, and equidistant to the listener, who is located at the so-called "sweet-spot," while facing the speakers.

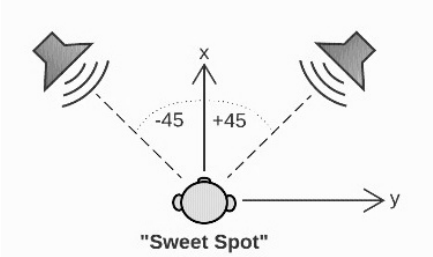


Figure 10.14: Speaker positioning and sweet spot.

Note that the range of panning (for stereo) is thus 90 degrees. However, it is practical to use radians instead of degrees. By convention, the left speaker is at 0 radians and the right speaker is at $\pi/2$ radians, giving us a panning range of $\theta \in [0; \pi/2]$, with the center position at $\theta = \pi/4$.

#### Linear Panning

The simplest panning strategy is to adjust the channel gains (volumes) linearly with inverse correlation.

> **Main idea**: for a stereo signal with gain 1, the gains of the left and right channels should sum to 1; i.e. $L(\theta) + R(\theta) = 1$.

This pdf for ICM students only - ebook
and paperback available from amazon.com

With the panning angle $\theta \in [0; \pi/2]$ we thus get the gain functions

$$L(\theta) = (\frac{\pi}{2} - \theta)\frac{1}{\frac{\pi}{2}} = (1 - \theta\frac{2}{\pi})$$

, and

$$R(\theta) = \theta\frac{1}{\frac{\pi}{2}} = \theta\frac{2}{\pi}$$

as plotted in Figure 10.15.

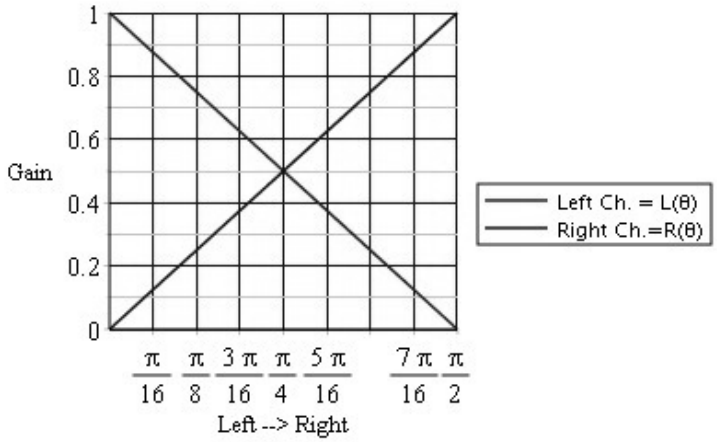


Figure 10.15: Linear panning.

A drawback of implementing panning in this way is that, even though the gains $L(\theta)$ and $R(\theta)$ always sum to 1, the loudness of the signal is still affected. Linear panning creates a "hole-in-the-middle" effect, such that the signal is softer at the middle than at the side-positions. At the center position, where $\theta = \pi/4$, we have $L(\pi/4) + R(\pi/4) = 1$. However, when a signal is panned to the center, the amplitudes coming from two speakers will typically not sum (unless they are perfectly in phase). In general, due to reflections and phase differences, the power is additive.[2] If we add power (proportional to the squared amplitude) we get $L^2(\pi/4) + R^2(\pi/4) = 0.5^2 + 0.5^2 = 0.5$. Expressed in dB, measuring down from the maximum gain (amplitude) of 1 (0 dB), we get $10\log_{10}(0.5)\text{dB} = -3\text{dB}$. Thus, when a signal is panned to the middle, it sounds 3 dB quieter than when panned fully left or right!

### Constant Power Panning

One way of dealing with the "hole-in-the-middle" effect is to use *constant power panning*. Basically, we change the linear functions for $L(\theta)$ and $R(\theta)$ to the sine and cosine functions, letting $L(\theta) = \cos(\theta)$ and $R(\theta) = \sin(\theta)$.

> **Main idea**: power is proportional to the squared amplitude, and $\cos^2 + \sin^2 = 1$.

Thus, $L(\theta) = \cos(\theta)$ and $R(\theta) = \sin(\theta)$ yields constant power panning.

As seen in the Figure 10.16, this boosts the center, giving us a gain of $cos(\pi/4) = \sin(\pi/4) = 0.71$ as opposed to the 0.5 center gain we saw with linear panning.

[2] This also relates to the law of conservation of energy.

This pdf for ICM students only - ebook
and paperback available from amazon.com

Now, the power of the signal at the center is $L^2(\pi/4)+R^2(\pi/4) = \cos^2(\pi/4)+\sin^2(\pi/4) = 1$, which is $10\log_{10}(\frac{1}{1})$dB $= 0$dB. The per-channel attenuation is $20\log_{10}(0.71) = -3$ dB. Thus, the center pan position is boosted by 3 dB[3] compared to linear panning, and the total power at every pan position is 0 dB.

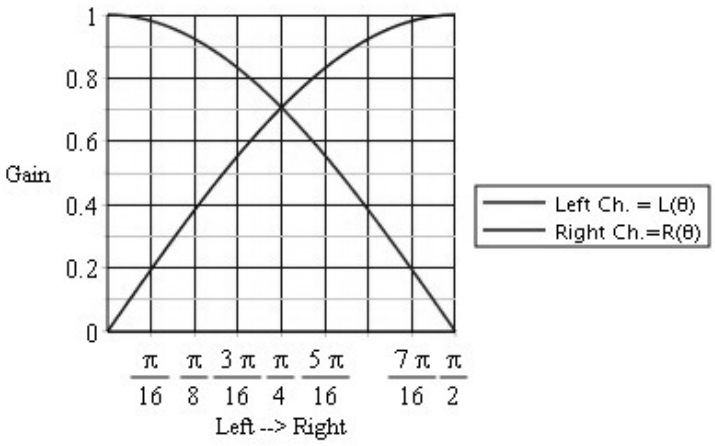


Figure 10.16: Constant power panning.

**-4.5 dB Pan Law (the compromise)**

What if our statement that power is additive is wrong? If amplitudes add, then the center pan position will get a 3 dB boost. This can happen if you convert stereo to mono by adding the left and right channels (so anything panned to the center is now added perfectly in phase), or if you sit in exactly the right spot for left and right channels to add in phase.[4] The idea behind the -4.5 dB law is to split the difference between constant power and linear panning – a kind of compromise between the two. This is achieved by simply taking the square root of the product of the two laws, thus we have $L(\theta) = \sqrt{(\frac{\pi}{2}-\theta)\frac{2}{\pi}\cos(\theta)}$, and $R(\theta) = \sqrt{\theta\frac{2}{\pi}\sin(\theta)}$; as plotted in Figure 10.17.

As we can see on the plot, the center gain is now at 0.59, and hence the per-channel attenuation is $10\log_{10}(\frac{.59^2}{1^2})$dB $= -4.5$dB, which is exactly in between that for the previous two laws. The power of the signal at the center is now $L^2(\pi/4)+R^2(\pi/4) = .59^2+.59^2 = 0.71$, corresponding to $10\log_{10}(\frac{.59^2+.59^2}{1^2})$dB $= -1.5$dB. If amplitudes are additive when stereo is converted to mono, the center pan signal is boosted by 1.5 dB. When signals are panned to the center and not heard in phase, the center pan signal is attenuated by 1.5 dB.

[3] In fact, the exact number is not 0.71 but $\sqrt{2}$, which in dB is approximately 3.0103. This is so close to 3 that even formal texts refer to this number as "3 dB," and a linear factor of 2 is often called "6 dB" instead of 6.0206. This is similar to calling 1024 bytes "one kilobyte." If you ever wonder how a factor of 2 got to be exactly 6 dB, well, it didn't!

[4] This still does not violate the law of conservation of energy. In a room, even mono signals cannot stay in phase everywhere, so the total power is conserved even if there are some "hot spots."

This pdf for ICM students only - ebook
and paperback available from amazon.com

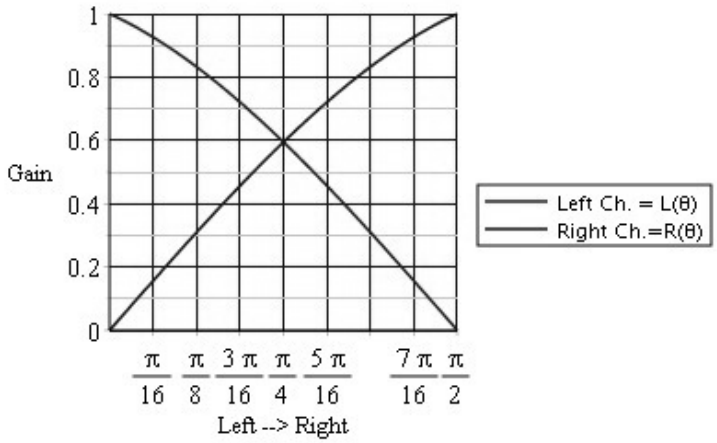


Figure 10.17: The -4.5 dB pan law.

**Which Pan Law to Use When?**

According to a number of sources (and as an interesting historic aside), back in the 1930's, the Disney corporation tested different pan laws on human subjects, and found that constant power panning was preferred by the listeners. Then, in the 1950's, the BBC conducted a similar experiment and concluded that the -4.5 dB compromise was the better pan law. Now, we know next to nothing about the details of those experiments, but their results might make sense if we assume that Disney's focus was on movie theaters and the BBC's was on TV audiences. Let's elaborate on that. It all depends on the listening situation. That is, how are the speakers placed, what is the size of the room, where is the listener placed, and can we expect the listener to stay in the same place? If the speakers are placed close to each other in a small room (with very little reverb) or if the listener is in the sweet spot, then one can reasonably expect that the phases of the signals from the two speakers will add up constructively (at least at lower frequencies, acting pretty much as one single speaker (at least at lower frequencies); as depicted in Figure 10.18.

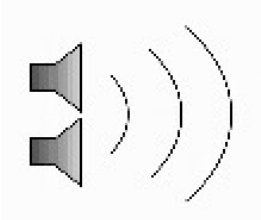

Figure 10.18: Two speakers in phase acting as one.

In such a case where signals are in phase so that the left and right amplitudes sum, the sounds that are panned to the center will experience up to 3 dB of boost with constant power panning but a maximum of only 1.5 dB boost using the -4.5 dB compromise. Thus, the -4.5 dB rule might give more equal loudness panning. It could also be that the BBC considered mono TV sets where the left and right channels are added perfectly in phase. In this case, the -4.5 dB compromise gives a 1.5 dB boost to center-panned signals vs. a 3 dB boost with constant power panning. (Recall that linear panning is ideal for mono reproduction because the

This pdf for ICM students only - ebook
and paperback available from amazon.com

center-panned signals are not boosted at all. However, linear panning results in the "hole-in-the-middle" problem for stereo.)

On the other hand, if the speakers are placed far from each other in a big room, then the phases will not add up constructively; as seen in Figure 10.19.

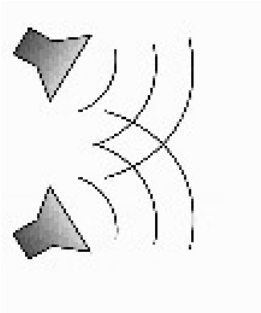

Figure 10.19: Two speakers out of phase; phases do not add constructively.

Furthermore, in this case, the listeners are probably not always placed at the sweet spot – there are probably multiple listeners placed at different distances and angles to the speakers (as e.g. in a movie theater). One would expect constant power panning to produce more uniform loudness at all panning positions in this situation.

Given the variables of listener position, speaker placement, and possible stereo-to-mono conversion, there is no universal solution that optimizes panning. Both constant power panning and the -4.5 dB compromise are widely used, and both are preferred over linear panning.

### Panning and Room Simulation

A more sophisticated approach to panning is to consider that in stereo recording, a left-facing microphone signal is very different from that of a right-facing microphone. Signal differences have to do with room reflections and the source location. A sound source at stage left will undergo different modifications getting to each microphone, and similarly for the source at stage right. Thus there are *four* different paths from sources to microphones, and that is considering only two source locations! Some panning systems take these paths into consideration or even integrate panning with reverberation simulation.

### Panning in Nyquist

In Nyquist, panning splits a monophonic signal (single channel) into stereo outputs (two channels; recall that multiple channel signals are represented using arrays of sounds), and the degree of left or right of that signal can be controlled by either a fixed parameter or a variable control envelope:

`pan(`*sound*`,` *where*`)`

The `pan` function pans *sound* (a behavior) according to *where* (another behavior or a number). *Sound* must be monophonic. The *where* parameter should range from 0 to 1, where 0 means pan completely left, and 1 means pan completely right. For intermediate values, the sound is scaled *linearly* between left and right.

This pdf for ICM students only - ebook
and paperback available from amazon.com

### 10.3.7 Compression/Limiting

A Compression/Limiting effect refers to automatic gain control, which reduces the dynamic range of a signal. Do not confuse dynamics compression with data compression such as producing an MP3 file. When you have a signal that ranges from very soft to very loud, you might like to boost the soft part. Alternatively, when you have a narrow dynamic range, you can expand it to make soft sounds much quieter and louder sounds even louder.

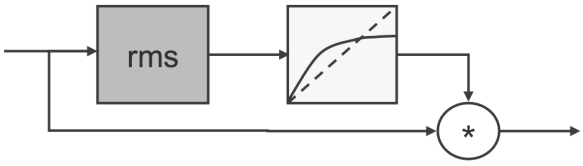


Figure 10.20: Compression/Limiting effect. The input signal is analyzed to obtain the RMS (average) amplitude. The amplitude is mapped to obtain a gain, and the signal is multiplied by the gain. The solid line shown in the mapping (at right) is typical, indicating that soft sounds are boosted, but as the input becomes increasingly loud, the gain is less, reducing the overall dynamic range (i.e. the variation in amplitude is compressed.)

The basic algorithm for compression is shown in Figure 10.20. The compressor detects the signal level with a Root Mean Square (RMS) detector[5] and uses table-lookup to determine how much gain to place on the original signal at that point. The implementation in Nyquist is provided by the Nyquist Extension named `compress`, and there are two useful functions in the extension. The first one is `compress`, which compresses *input* using *map*, a compression curve probably generated by `compress-map`.[6] Adjustments in gain have the given *rise-time* and *fall-time*. *lookahead* tells how far ahead to look at the signal, and is *rise-time* by default. Another function is `agc`, an automatic gain control applied to *input*. The maximum gain in dB is *range*. Peaks are attenuated to 1.0, and gain is controlled with the given *rise-time* and *fall-time*. The look-ahead time default is *rise-time*.

`compress(`*input*`,` *map*`,` *rise-time*`,` *fall-time*`,` *lookahead*`)`

`agc(`*input*`,` *range*`,` *rise-time*`,` *fall-time*`,` *lookahead*`)`

### 10.3.8 Reverse

Reverse is simply playing a sound backwards. In Nyquist, the reverse functions can either reverse a sound or a file, and both are part of the Nyquist extension named `reverse`. If you reverse a file, Nyquist reads blocks of samples from the

[5]The RMS analysis consists of squaring the signal, which converts each sample from positive or negative amplitude to a positive measure of power, then taking the mean of a set of consecutive power samples perhaps 10 to 50 ms in duration, and finally taking the square root of this "mean power" to get an amplitude.

[6]`compress-map(`*compress-ratio*`,` *compress-threshold*`,` *expand-ratio*`,` *expand-threshold*`,` *limit*`,` *transition*`,` *verbose*`)` constructs a map for the compress function. The map consists of two parts: a compression part and an expansion part. The intended use is to compress everything above compress-threshold by *compress-ratio*, and to downward expand everything below *expand-threshold* by *expand-ratio*. Thresholds are in dB and ratios are dB-per-dB. 0 dB corresponds to a peak amplitude of 1.0 or RMS amplitude of 0.7. If the input goes above 0 dB, the output can optionally be limited by setting `limit:` (a keyword parameter) to `T`. This effectively changes the compression ratio to infinity at 0 dB. If `limit:` is `nil` (the default), then the *compression-ratio* continues to apply above 0 dB.

This pdf for ICM students only - ebook
and paperback available from amazon.com

file and reverses them, one-at-a-time. Using this method, Nyquist can reverse very long sounds without using much memory. See `s-read-reverse` in the *Nyquist Reference Manual* for details.

To reverse a sound, Nyquist must evaluate the whole sound in memory, which requires 4 bytes per sample plus some overhead. The function `s-reverse(`*sound*`)` reverses *sound*, which must be shorter than `*max-reverse-samples*` (currently initialized to 25 million samples). This function does sample-by-sample processing without an efficiently compiled unit generator, so do not be surprised if it calls the garbage collector a lot and runs slowly. The result starts at the starting time given by the current environment (not necessarily the starting time of *sound*). If *sound* has multiple channels, a multiple channel, reversed sound is returned.

### 10.3.9 Sample-Rate Conversion

Nyquist has high-quality interpolation to alter sample rates. There are a lot of sample rate conversions going on in Nyquist behind the scenes: Nyquist implicitly changes the sample rate of control functions such as produced by `env` and `lfo` from their default sample rate which is 1/20 of the audio sample rate. When you multiply an envelope by an audio rate signal, Nyquist *linearly* interpolates the control function. This is reasonable with most control functions because, if they are changing slowly, linear interpolation will be a good approximation of higher-quality signal reconstruction, and linear interpolation is fast. However, if you want to change the sample rate of audio, for example if you read a file with a 48 kHz sample rate and you want the rate to be 44.1 kHz, then you should use high-quality sample-rate conversion to avoid distortion and aliasing that can arise from linear interpolation.

The algorithm for high-quality sample-rate conversion in Nyquist is a digital low pass filter followed by digital reconstruction using *sinc* interpolation. There are two useful conversion functions:

`force-srate(`*srate*`, `*sound*`)`

returns a sound which is up- or down-sampled to *srate*. Interpolation is linear, and no pre-filtering is applied in the down-sample case, so aliasing may occur.

`resample(`*snd*`, `*rate*`)`

Performs high-quality interpolation to reconstruct the signal at the new sample rate. The result is scaled by 0.95 to reduce problems with clipping. (Why? Interestingly, an interpolated signal can reconstruct peaks that exceed the amplitude of the the original samples.)

Nyquist also has a variable sample-rate function:

`sound-warp(`*warp-fn*`, `*signal*`, `*wrate*`)`

applies a warp function *warp-fn* to *signal* using function composition. If the optional parameter *wrate* is omitted or NIL, linear interpolation is used. Otherwise, high-quality sample interpolation is used, and the result is scaled by 0.95 to reduce problems with clipping. Here, *warp-fn* is a mapping from score (logical) time to real time, and *signal* is a function from score time to real values. The result is a function from real time to real values at a sample rate of `*sound-srate*`. See the *Nyquist Reference Manual* for details about *wrate*.

To perform high-quality stretching by a fixed ratio, as opposed to a variable ratio allowed in `sound-warp`, use `scale-srate` to stretch or shrink the sound, and then `resample` to restore the original sample rate.

This pdf for ICM students only - ebook
and paperback available from amazon.com

### 10.3.10 Sample Size Conversion (Quantization)

Nyquist allows you to simulate different sample sizes using the unit generator `quantize`:

`quantize(`*sound*`, `*steps*`)`

This unit generator quantizes *sound* as follows: *sound* is multiplied by *steps* and rounded to the nearest integer. The result is then divided by *steps*. For example, if *steps* is 127, then a signal that ranges from -1 to +1 will be quantized to 255 levels (127 less than zero, 127 greater than zero, and zero itself). This would match the quantization Nyquist performs when writing a signal to an 8-bit audio file. The *sound* may be multi-channel.

### 10.3.11 Reverberation

A reverberation effect simulates playing a sound in a room or concert hall. Typical enclosed spaces produce many reflections from walls, floor, ceiling, chairs, balconies, etc. The number of reflections increases exponentially with time due to secondary, tertiary, and additional reflections, and also because sound is following paths in all directions.

Typically, reverberation is modeled in two parts:

- Early reflections, e.g. sounds bouncing off one wall before reaching the listener, are modeled by discrete delays.
- Late reflections become very dense and diffuse and are modeled using a network of all-pass and feedback-delay filters.

Reverberation often uses a low-pass filter in the late reflection model because high frequencies are absorbed by air and room surfaces.

The rate of decay of reverberation is described by RT60, the time to decay to -60 dB relative to the peak amplitude. (-60 dB is about 1/1000 in amplitude.) Typical values of RT60 are around 1.5 to 3 s, but much longer times are easy to create digitally and can be very interesting.

In Nyquist, the `reverb` function provides a simple reverberator. You will probably want to mix the reverberated signal with some "dry" original signal, so you might like this function:

```
function reverb-mix(s, rt, wet)
  return s * (1 - wet) + reverb(s, rt) * wet
```

Nyquist also has some reverberators from the Synthesis Tool Kit: `nrev` (similar to Nyquist's reverb), `jcrev` (ported from an implementation by John Chowning), and `prcrev` (created by Perry Cook). See the *Nyquist Reference Manual* for details.

#### Convolution-based Reverberators

Reverberators can be seen as very big filters with long irregular impulse responses. Many modern reverberators measure the impulse response of a real room or concert hall and apply the impulse response to an input signal using convolution (recall that filtering is equivalent to multiplication in the frequency domain, and that is what convolution does).

With stereo signals, a traditional approach is to mix stereo to mono, compute reverberation, then add the mono reverberation signal to both left and right

This pdf for ICM students only - ebook
and paperback available from amazon.com

channels. In more modern reverberation implementations, convolution-based reverberators use 4 impulse responses because the input and output are stereo. There is an impulse response representing how the stage-left (left input) signal reaches the left channel or left ear (left output), the stage-left signal to the right channel or right ear, stage-right to the left channel, and stage-right to the right channel.

Nyquist has a `convolve` function to convolve two sounds. There is no Nyquist library of impulse responses for reverberation, but see `www.openairlib.net/`, `oramics.github.io/sampled/IR/Voxengo/` and other sources. Convolution with different sounds (even if they are not room responses) is an interesting effect for creating new sounds.

### 10.3.12 Summary

Many audio effects are available in Nyquist. Audio effects are crucial in modern music production and offer a range of creative possibilities. Synthesized sounds can be greatly enhanced through audio effects including filters, chorus, and delay. Effects can be modulated to add additional interest.

Panning algorithms are surprisingly non-obvious, largely due to the "hole-in-the-middle" effect and the desire to minimize the problem. The -4.5 dB panning law seems to be a reasonable choice unless you know more about the listening conditions.

Reverberation is the effect of millions of "echoes" caused by reflections off of walls and other surfaces when sound is created in a room or reflective space. Reverberation echos generally become denser with greater delay, and the sound of reverberation generally has an approximately exponential decay.