

Using Precoded Modules to Produce and Manipulate Audio

Mark Berry

January 23, 2004

Abstract

Producing sound on a computer involves many steps. Aligning modules to perform these steps involves a lot of programming. For example, input/output programs that port computer files to the program and the output to the speakers must be aligned with software that alters volume, pitch, panning, equalization, etc.

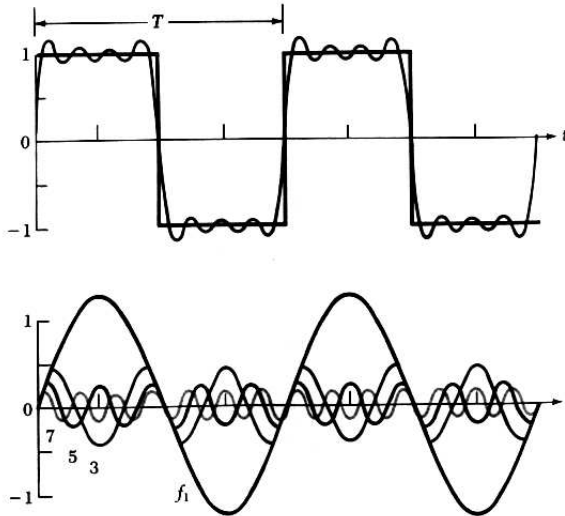
I'm (so far) using two programs to perform this function: CCRMA and XMP. The former is a simple developer-based music and MIDI sound system: it allows you to alter the speed and volume of sound, and contains algorithms to play the sound through the speakers. XMP is a MOD player, which basically converts WAV or similar files into music by controlling various aspects of the tone. Though I have little interest in coding MODs to produce music, I can integrate these aspects of the MOD software to alter music outputted by the CCRMA code.

1 Sound Production

Sound is stored in Fourier Transforms. In order to accommodate the input/output requirements of the computer, the sound must be converted into this format because of the nature of the technology. This means that when the sound is buffered for output, it is in this format and must be decomposed in order to effectively manipulate it. All sound-producing software does this. In order to convert the buffer sound into the manipulatable sine form we are used to seeing, we must use the following formula:

$$F(k) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i k x} dx.$$

We would use both the positive and negative exponent for e in order to find both the forward and the inverse forms of the sine wave. Of course, no computer could possibly handle this equation all the way to infinity, but we approach it as high as possible.



This is an example of how a Fourier transform works. By continuously adding sine waves together using the above formula, a mathematical representation of an actual sound wave can be produced. The fidelity of the transform depends on how many iterations are performed. In this case, the square wave in the top figure is simulated by adding up all the sine waves in the bottom figure.

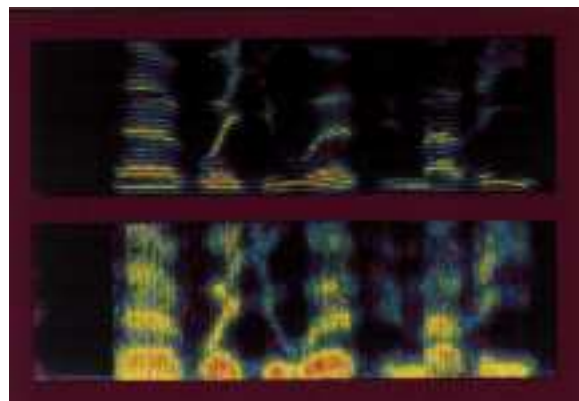
There are only four iterations in this example; if there had been more, the sound produced by the computer would closer approximate the actual sound.

2 Equalization

An equalizer is a program or device that alters sound input in some way to change output. The most common equalizer alters the volume of specific bands of frequency. When audio is loaded, it is placed in a buffer, where algorithms can be applied to it. Once the

computer has converted the buffer from a Fourier transform, it can use what's called a Digital Bandpass Filter (DBF), which blocks or amplifies specified frequencies (itches) of audio. This is a complicated mathematical equation: you must set the frequency (pitch), gain (the amplitude, usually from -15dB to +15dB) and steepness. Steepness is the way that an amplification of a certain frequency alters the amplitudes of the frequencies nearby it. For example, if the bass were raised, there would be a gradation between the loud bass and the soft treble.

Finally, an important effect of an equalizer is the preamplifier. This raises or lowers the entire sound output based on certain conditions: if the song goes from very loud to very soft, it might dynamically lower the volume of the song or of certain frequencies in order to maintain an even sound level. It is also important in order to avoid overloading the speakers.



The graphic above is an example of a sound produced across bands of audio, called a "voiceprint." The x-axis plots time, and the y-axis plots the amplitude of frequencies of the tone. Notice that the bass is heavy;

an expert would use this graph to raise the pitch on an equalizer.

3 XMP and CCRMA

XMP, or Extended MOD Player, is an open-source program that converts user-composed MOD files into sound output. What this means is that it takes a few WAV files and manipulates them according to instructions coded into the MOD. There are thousands of audio-manipulation abilities that the player is capable of, such as crescendos, panning, bass amplification, speed/pitch alterations, and so on. Since XMP is open-source, the code is freely available, so it can be added to my music player.

CCRMA is a code set programmed by Stanford. This set is developer-oriented, and includes functions such as a player and an input-output codec. These are separate files that can be added to a program at will, and they can be interspersed with other programming. This is very useful for programming an equalizer, because it can be combined with XMP's WAV-slowing and pitch-varying codecs in order to complete a player that equalizes.

Of course, combining CCRMA and XMP is extremely difficult. They were written by completely different people for different reasons, and so they must be carefully accommodated to align each program's input and output.