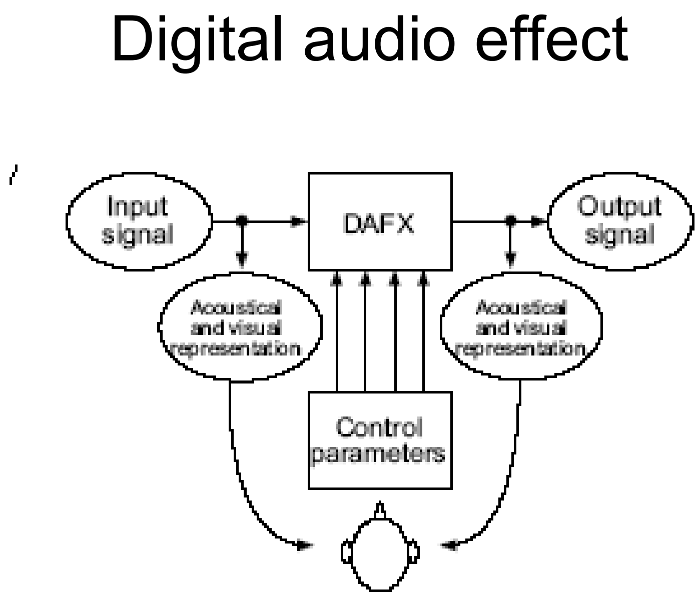

Figure 1: Digital Audio Effect and its control

Digital Audio Effects

Basically any processing system ("box") that takes as input an audio signal which is modified according to some sound control parameters and delivers an output audio signal. The input and output signals are monitored by loudspeakers or headphones as well as possibly visual representations of the signal such as waveforms and spectrograms. Typically a human (musician, recording engineer, student) modifies the control parameters in order to achieve a particular effect on the sound. Figure 1 shows this process.

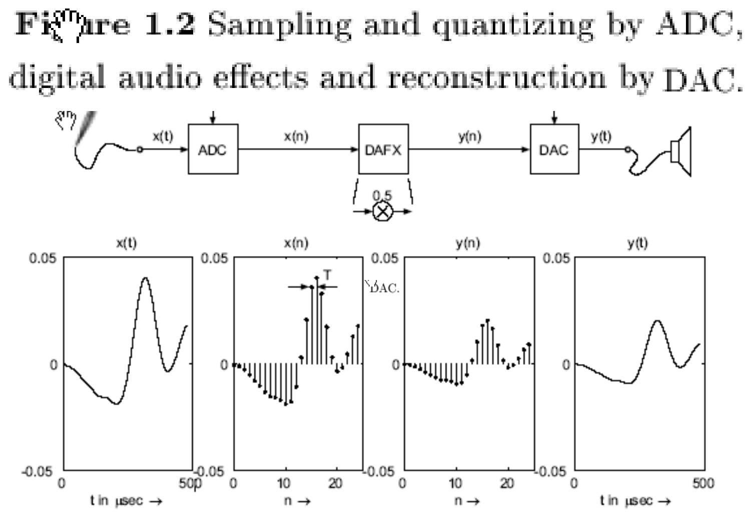

Figure 2: Signal Flow Diagram

Visual representations of digital systems is achieved by functional block diagram representations or signal flow graphs. Figure 2 shows such a diagram for a simple system for sampling and quantization by ADC, a simple gain digital audio effect and reconstruction by a DAC. As we will see later in this course similar dataflow representations can be used as the basic for designing software architectures for audio processing.

An important consideration when implementing Digital Audio Effects in software is whether the processing is sample-by-sample based or block-based (a block or window is a range of N samples held in a memory buffer so that the audio effect can operate on them).

Sinusoids and Phasors

Some amazing mathematical ideas that we will explore in this class:

- Any sound you hear can be broken down to a sum of sine waves
- Any sound you hear can be broken down into a file of bits - on/off positions of switches

These ideas will enable us to store, generate and manipulate sounds on a computer in many interesting ways. The main challenge we will face in this class is to understand how arbitrary sounds are affected when they pass through audio signal processing systems. One of the most fundamental ideas that will help us address this challenge is to try to represent a complex audio signal as a mixture of simpler "building blocks". In order to be useful for understanding audio signal processing systems these "building blocks" it is important for them to have the following characteristics:

- Understanding how a particular family of audio signal processing algorithms affects the simpler "building blocks" should be easier than understanding how a complex arbitrary sound is affected.
- It should be possible to generalize knowledge from how the simpler "building blocks" are affected to how a complex arbitrary sound is affected.
- The "building blocks" should be somehow regular and easy to parametrize

We will start our exploration by investigating the most fundamental audio signal and arguably the most useful "building block" for understanding audio signal processing **a sinusoid**.

$$x(t) = sin(\omega t + \phi) \tag{1}$$

Some interesting observations:

- Notice that $sin(\omega t)$ and $cos(\omega t)$ are just shifted version of ach other

$$cos(\omega t) = sin(\omega t + \pi/2) \tag{2}$$

- The time derivatives of a sinusoid are also sinusoids

$$\frac{d}{dt} sin(\omega t) = \omega cos(\omega t) \quad (3)$$

$$\frac{d}{dt} cos(\omega t) = -\omega sin(\omega t) \quad (4)$$

  That means that any sinusoid satisfies the equation of simple harmonic motion (i.e a function x(t) which is proportional to its second derivative). Simple harmonic motion represents a variety of idealized physical motions including: a tuning fork (small angle approximation to mass in a pendulum), mass on a spring, uniform circular motion.

- The relative phase angle ϕ is arbitrary and is determined by the choice of time origin. More mathematically the set of all sinusoids at a fixed frequency is closed under the operation of time shift. In this sense the "shape" of a sinusoid is the same regardless of when we observe it.

- Another very important closure property is that adding two sinusoids of the same frequency, but not necessarily with the same phases of amplitudes, produces another sinusoid with the same frequency (of different amplitude and phase). This property is not that obvious. See figure 3.

  The brute force way to show this would be to start with the sum

$$\alpha_1 cos(\omega t + \phi_1) + \alpha_2 cos(\omega t + \phi_2) \quad (5)$$

  where $\alpha_1, \alpha_2, \phi_1, \phi_2$ are arbitrary constants, and do some messy algebra. It's a good exercise to try to prove this. You might find the following trinometric identities useful:

$$cos(\theta + \phi) = cos(\theta)cos(\phi) - sin(\theta)sin(\phi) \quad (6)$$

$$sin(\theta + \phi) = sin(\theta)cos(\phi) + cos(\theta)sin(\phi) \quad (7)$$

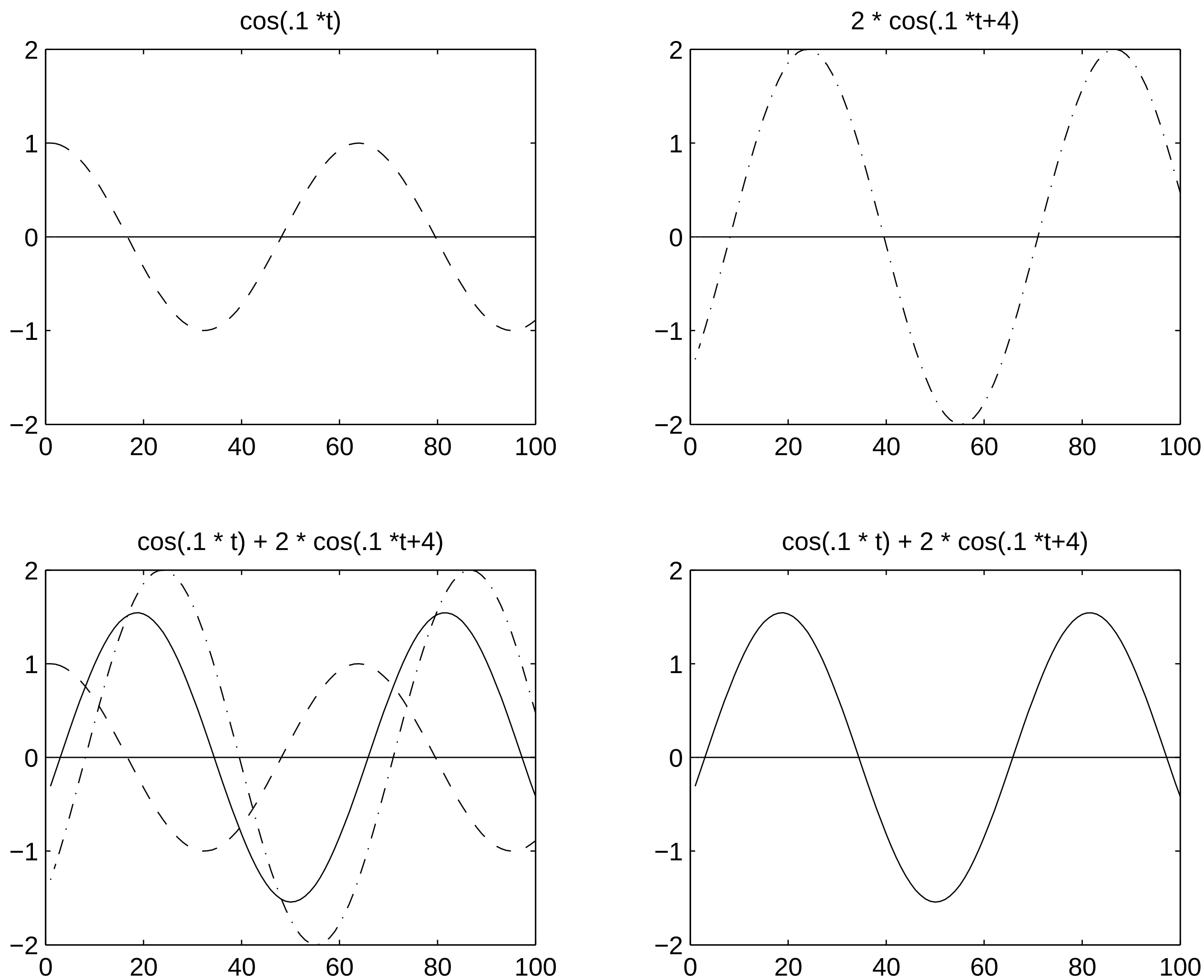

Figure 3: Adding sinusoids of the same frequency

The MATLAB code for generating Figure 3 is provided here:

```
t = 0:100;

% Create a sinusoid a
a = cos(0.1 * t);
zrs = zeros(size(a));

% Plot sinusoid a
subplot(2,2,1), plot(a, 'red--');
hold; plot(zrs, 'black');
axis([0 100 -2 2]);
title('cos(.1 *t)');

% Create a second sinusoid b
% with same frequency but different amplitude and phase
b = 2 * cos(.1 * t + 4);

% Plot sinusoid b
subplot(2,2,2); plot(b, 'blue-.');
hold; plot(zrs, 'black');
axis([0 100 -2 2]);
title('2 * cos(.1 *t+4)');
```

Plot sum of sinusoids by itself

```
subplot(2,2,4);
plot(a+b, 'black');
hold;
plot(zrs, 'black');
axis([0 100 -2 2]);
title('cos(.1 * t) + 2 * cos(.1 *t+4)');

subplot(2,2,3); plot(a, 'red--');
hold; plot(b, 'blue-.');
plot(a+b, 'black');
plot(zrs, 'black');
axis([0 100 -2 2]);
title('cos(.1 * t) + 2 * cos(.1 *t+4)');
```

Measures of Amplitude

One of the most fundamental properties of any digital audio signal is its amplitude. Unfortunately the term "amplitude" is used in many different ways. For example amplitude of a sample, sinusoid or amplitude of a window (a fixed range of samples of the signal). We will consider a window starting at sample M of length N of an audio signal x[n]:

$$x[M], x[M+1], ..., x[M+N-1] \tag{8}$$

Frequently used amplitude measures are the *peak* amplitude:

$$PA\mathbf{X} = max x[n], n = M, ...M+N-1 \tag{9}$$

and the Root Mean Square (RMS) amplitude:

$$RMS\mathbf{X} = \sqrt{\frac{1}{N}\sum_{n=M}^{M+N-1} x[n]^2} \tag{10}$$

Under reasonable conditions the peak amplitude of a sinusoid is α (the amplitude of the sinusoid) and the RMS amplitude $\alpha/\sqrt{2}$.

Decibels

Typically measures of amplitude (peak or RMS) are expressed in *decibels* which are logarithmic units. If α is the amplitude of a signal then we can definte the decibel (dB) level d as:

$$d = 20 * log_{10}(\alpha/\alpha_o) \tag{11}$$

where α_0 is a reference amplitude. Doubling amplitude corresponds to an increase of about 6.02 dB. Doubling power (the square of amplitude) corresponds to an increase of 3.01 dB.

In digital audio a convenient choice of reference, assuming the hardware has a maximum amplitude of 1.0 is:

$$\alpha_0 = 10^{-5} = 0.00001 \tag{12}$$

so that the maximum possible amplitude is 100 dB and 0 dB is almost inaudible.

The dynamic range of hearing for humans is also approximately 100 dB. It is important to realize that the digital amplitude values are relative and unitless (you can turn the loudest sound that can be represented digitally "off" by just lower the volume knob) whereas the amplitude values related to human hearing are absolute (for example a loud rock concert is about 120 dB and is probably bad for the long term health of your ears)

Homework

- Prove that the sum of two sinusoids of the same frequency with different amplitude and phases is also a sinusoid of the same frequency using algebra and trigonometric identities.
- Read chapter 1 of the DAFX book. Bring any questions you have in class on Thursday
- Refamiliarize yourself with MATLAB using the code in Chapter 1
- Play with Audacity