Lab02: Sampling and Aliasing

EECS20: Introduction to Real-Time Digital Systems

©1996 Regents of the University of California.

By K. H. Chiang, William T. Huang, Brian L. Evans.

URL: `http://www-inst.eecs.berkeley.edu/~ee20`

News: `ucb.class.ee20`

Assigned: 5 Feb 97, Checkoff: 12 Feb 97, Writeup Due: 14 Feb 97

1 Introduction

In lecture, we saw that a signal could be thought of as a measurement of some physical phenomenon. For example, a microphone changes a sound signal to an electrical signal by converting the variations in the pressure of the air to variations in the voltage on a conducting wire. Both the sound and electrical signals continuously vary with time and are called analog signals. In order to represent an analog signal on a digital computer, we approximate the analog signal by taking a finite number of samples [sampling] and then assigning a value to each sample [quantization]. In this laboratory, we will investigate sampling and aliasing in the context of tones.

2 Sampling

We have all seen signals in various guises:

- the height of a vehicle from the road surface as it is driven across a pothole-strewn road.
- the intensity of sound from your speakers when you're playing your favorite CD.
- the temperature in your room as a function of the time of day.

A one-dimensional signal is a function of a single variable. This variable is usually time, as it is in the four examples above. It need not be time, however; in image processing, the index variable is spatial– that is, we could be examining at intensity on a two-dimensional image as a function of where we're currently looking. For now though, let's stick with time as the index variable.

For the purposes of this course, we will be largely dealing with discrete time (DT) signals. DT signals are defined for a subset of points in time; usually this subset contains points evenly spaced in time. Examples of DT signals include:

- a per-minute count of drops falling from a faucet in that minute.
- the voltage at some point in a circuit, sampled at half-second intervals.
- the closing value of the New York Stock Exchange. Note that if the signal were graphed against the days of the week, it would be undefined on Saturdays and Sundays.

In contrast to a DT signal, a continuous time (CT) signal is defined for all points in time. However, when we try to represent a CT signal in a computer, we run into a problem: we can assign a given signal a value at any point in time, but where should we choose the *next* point in time in relation to the current one? Between any two points in time, there are an infinite number of points. So in a digital computer we represent signals as discrete-time signals.[1]

What are the differences between CT and DT signals? A fundamental difference is what happens to sinusoids. Consider a CT sine wave $x_1(t)$ with frequency f_1:

$$x_1(t) = \sin 2\pi f_1 t$$

Now consider a DT signal which is the sampled version of the CT sine wave. What does it mean to sample a CT signal? It means that at evenly spaced points in time, we can think of closing a switch and recording the value of the signal at those points in time. If the sampling period [the time between successive samples] is T, then we have a new signal:

$$x_1[n] = x_1(nT) = \sin 2\pi f_1 nT$$

Note the change of variable from t to n. If we sample with, say, $T = 0.5$s, the value of the signal at $t = 1.0$s would be the same as the value of the signal at $n = 2$. Note also that n is an integer.

If the sampling period is T, the sampling frequency f_s is then $\frac{1}{T}$.

Why would one want to bother with sampling a sinusoid in the first place though? The key fact is this: *sound is pressure variation.* One of the most basic sounds, a pure tone, is generated through a sinusoidal variation in pressure.

3 Aliasing

What are the differences between CT and DT signals? A fundamental difference is what happens to sinusoids. Consider a CT sine wave $x_1(t)$ with frequency f_1:

$$x_1(t) = \cos 2\pi f_1 t$$

Now consider a CT sine wave $x_2(t)$ with frequency f_2:

$$x_2(t) = \cos 2\pi f_2 t$$

If $f_1 \neq f_2$, it's obvious that the two sinusoids are different. How dull and boring.

Let's take a look at DT. In particular, let's choose the frequencies such that $f_2 = f_1 + kf_s$, where k is an integer. $x_1[n]$ is:

$$x_1[n] = \cos 2\pi f_1 nT$$

[1] Analog computers are well-suited for handling continuous-time signals. An example is the automatic transmission in older cars– gears, hydraulics, and other hydromechanical whatnot are used determine when to shift from one gear to another.

What about $x_2[n]$?

$$\begin{aligned} x_2[n] &= \cos 2\pi f_2 nT \\ &= \cos 2\pi (f_1 + kf_s)nT \\ &= \cos(2\pi f_1 nT + 2\pi k\frac{1}{T}nT) \\ &= \cos(2\pi f_1 nT + 2\pi kn) \\ &= \cos 2\pi f_1 nT \end{aligned}$$

Notice that you can't tell the difference between $x_1[n]$ and $x_2[n]$. This effect— when the sampled versions of two different CT signals look the same— is called *aliasing*. Note that the frequencies which get aliased are dependent on the sampling period.

For example, consider a 1 Hz sine wave sampled with a sampling frequency of 8 Hz, as in Figure **??**; the points at which the sine is sampled are indicated as circles. As predicted by our formula [$f_1 + kf_s = 1\text{Hz} + 8\text{Hz}$ for $k = 1$], a 9 Hz sine wave passes through those exact same points.

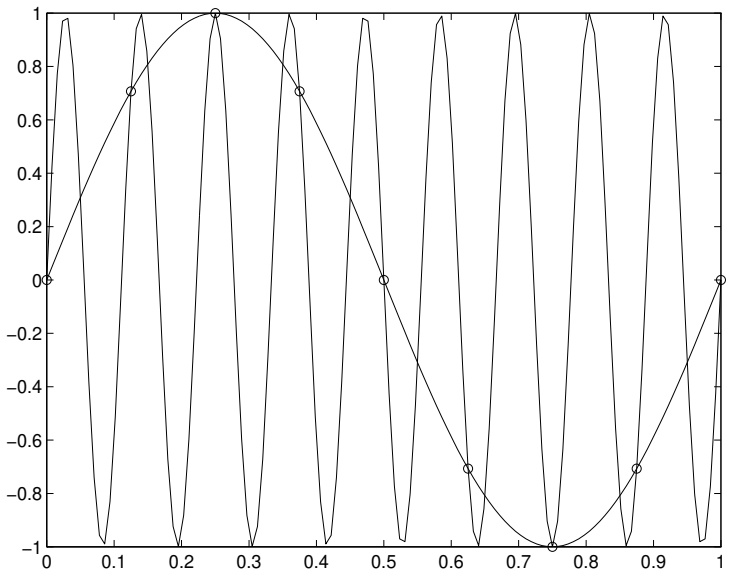

Figure 1: A 1 Hz sine wave sampled at 8 Hz looks like a 9 Hz sine wave. The circles indicate where the samples are taken.

4 Sampling and the Nyquist Rate

Another way to see aliasing is to sample a high frequency sinusoid at a rate which is too low to capture all the information in the CT signal.

It turns out that if you sample at a rate which is greater than twice the highest frequency in your CT signal, the frequency of the resulting DT signal will not alias to some other frequency. In other words:

$$f_s > 2f_{max}$$

The quantity $2f_{max}$ is referred to as the Nyquist rate.

5 Questions

1. Familiarize yourself with the following Matlab commands:

   - `sound`
   - `saxis`

   Note that the PCs in the lab have a `sound` command that takes a second parameter— the sampling frequency. Experiment with various values of this parameter. If you are using an HP, the sampling frequency is fixed to 8 kHz; you should go to the lab and play around with the PCs instead for this part.

2. Generate some sampled sinusoids using Matlab and the `sound` command. Vary the frequency of the sinusoid and the sampling period. Caution: if you set the frequency too low, you may potentially damage the speakers. [Why?]

3. Vary the amplitude and phase [watch out for the autoscaling of the `sound` command; check the `saxis` command to disable/reenable the autoscaling]. How can you make a sinusoid sound twice as loud? Can you hear the effect of varying phase?

4. Determine the lower and upper limits of your hearing. A good starting point is 440 Hz, which corresponds to middle A, as we noted in lecture.

5. Generate a 5 Hz sine wave, sampled at $f_s = 20$Hz. Generate some other sinusoids that alias to this 5 Hz sine wave.

6. Generate a 5 Hz sine wave, sampled at f_s equal to 2.5 Hz, 5 Hz, 10 Hz, and 20 Hz. Determine the Nyquist rate. Determine if aliasing has occurred in your sampled signal, and what those aliased frequencies are. Is sampling directly at the Nyquist rate a useful operation?

7. Generate and listen to a 440 Hz tone. Determine which frequencies alias to that tone and confirm this audibly. Determine the Nyquist rate.

8. In CT, low frequencies can be thought of as those frequencies which are close to 0 Hz and high frequencies as those that aren't. By plotting sampled sinusoids, determine the equivalent of high and low frequencies in DT. [Hint: both high and low frequencies will be multiples of some number.]

9. In lecture, we introduced the concept of discrete-time frequencies. If the sampling period is T, relate a given discrete-time frequency to the corresponding continuous-time frequency.