Introduction to Signals and Systems: Lecture Summary

In this lecture, we began with an overview of the course, covering key topics at a qualitative (rather than quantitative) level. We specifically talked about the following important concepts:

- What is meant by the term *signal*?

  Signals are measured quantities that are a function of one or more variables, and that carry some information content. Important examples cited included sound (e.g. speech, music), images and video. (See *Mathematica* notebook, section “Sample signals” for specific examples.) For many signals, the *independent* variable is time; for example, audio signals represent variations in pressure over time. Other signals are functions of one or more independent variables that are not time; an image, for example, is a signal that consists of intensity or color values that are a function of *x* and *y* position in the image.

- What is meant by the term *discrete-time*?

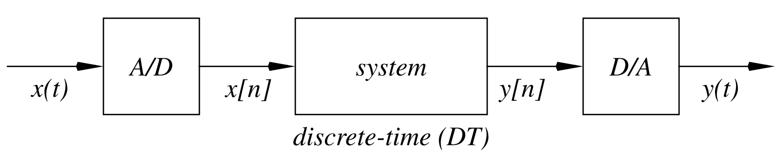
  Signals in the real world are functions of *continuous* time. For digital computers, however, we sample[1] real-world signals at (typically) regularly spaced intervals to convert them to *discrete-time* signals. Thus, unlike continuous-time signals, discrete-time signals can be represented as sequences of numbers. (See *Mathematica* notebook, section “Continuous-time vs. discrete-time” for examples of continuous-time to discrete-time conversion of signals through sampling.) As digital computers and technology have advanced over the past several decades, so has the importance of understanding and analyzing discrete-time signals. Operating on signals in the discrete-time domain as opposed to the continuous-time domain allows for much greater flexibility in the design of systems, since changes in design are typically a function of software, rather than hardware.

- What is meant by the term *system*?

  Systems operate or transform signals in some hopefully desirable way. The diagram below illustrates the concept of a system in block diagrams:

continuous-time (CT) *discrete-time (DT)*

  In the diagrams, *x(t)* and *y(t)* denote the input and output, respectively, of the continuous-time system on the left, while *x[n]* and *y[n]* denote the input and output, respectively, of the discrete-time system on the right. The diagram below illustrates how a typical system might convert a continuous-time signal *x(t)* to a discrete-time signal *x[n]* through sampling (*A/D*), operate or transform that signal through a discrete-time system to the discrete-time signal *y[n]*, and convert it back to a continuous-time signal *y(t) (D/A)*.



  Some important examples of systems include filters, encryption and control.

- *Time domain* vs. *frequency domain*:

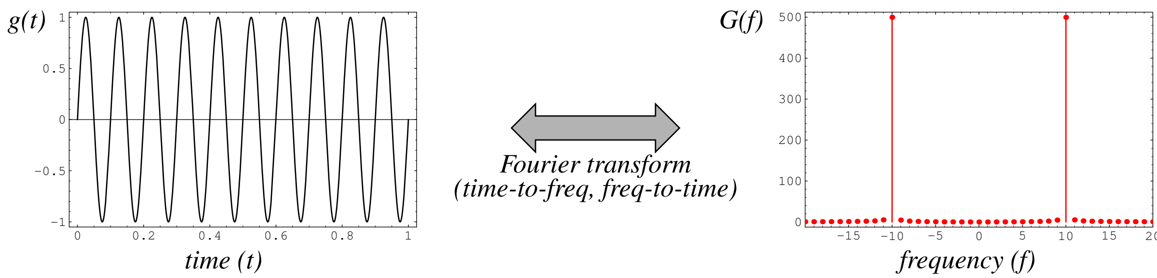
  We are used to viewing signals in the *time domain* — that is, as a function of time. however, there is another domain — namely, the *frequency domain*, that is critically important in understanding, analyzing and transforming signals. In lecture, we introduced the concept of *frequency*, and the *frequency domain*. As we will see in this

1. The process of sampling a continuous-time signal changes the signal, and it is important to understand how exactly the sampling process changes the information content of that signal. In due time, we will carefully look at the implications of sampling, and how choosing an appropriate sampling rate is critically important to preserving the information contained in a signal.

class, almost all signals can be represented as an infinite sum of sinusoids. (See *Mathematica* notebook, section "Listening to frequencies" to listen to sinusoids at different frequencies.) The frequency domain tells us the relative contribution of sinusoids of different frequencies to the overall signal. Consider the figure below, which illustrates a simple example of the transformation between the time and frequency domains. On the left, we plot the function $g(t)$:

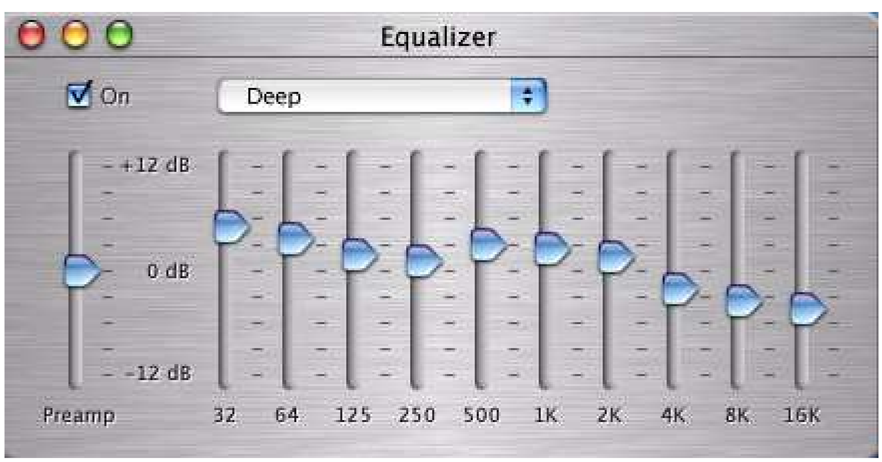
$$g(t) = \sin(2\pi f_c t) \tag{1}$$

where the frequency f_c is 10 Hertz (Hz, cycles/second). On the right, we plot the frequency representation $G(f)$ of that signal. Note how the frequency representation has two spikes at $f = \pm 10$ Hz, and is zero everywhere else.



(See *Mathematica* notebook, sections "Time domain $\Leftrightarrow$ Frequency domain" for further examples of time domain to frequency domain transformations.)

Once we understand that we can represent any time-varying signal in the frequency domain, the concept of an equalizer, as present in many audio systems and mp3 players begins to make sense. Consider for example the equalizer of the *iTunes* mp3 player pictured below:



Note the numbers at the bottom of each of the slider bars (e.g. 32, 64, 125, etc...). These represent frequency bands centered around 32Hz, 64Hz, 125Hz, ..., respectively. By increasing the slider value corresponding to one of the frequency bands, we are essentially saying, "transform the signal to amplify those frequency components of the signal (i.e. mp3 file)." Different combinations of gains achieve different effects, such as "Bass Boosting" (amplifying low-frequency values) or "Treble Boosting" (amplifying high-frequency values). This kind of signal modification would be very difficult to achieve if the designers of the equalizer did not have a very good understanding of the frequency domain, and the frequency representation of time-varying signals. (Note that the equalizer above is a very familiar example of a *system*.)

The mathematical techniques that allow us to transform a signal from the time domain to the frequency domain and back —namely, the Fourier transform — will be a key focus of this course.