Signals and LTI Systems

Eric Jonas (jonas@eecs.berkeley.edu)

June 15, 2015

1 What is a signal

A signal is a function of one or more independent variables, such as $x(t)$, where we generally let t range over $(-\infty, \infty)$. We often think of such signals as representing some observation of the world – the voltage on a wire, the air pressure at a microphone, or electromagnetic vibrations in the vacuum.

We will work with signals in continuous time, and represent them by $x(t)$, $t \in \mathbb{R}$. We will also work with signals in discrete time, $x[n]$, $n \in \mathbb{Z}$. Note that the behavior of these two different classes of signals can be both subtle and profound, and transitioning between one and the other implactes some of the most classic results in signal processing, such as the Shannon/Nyquist sampling theorem. However, a large number of the properties that apply to discrete signals apply to continuous-time signals and vice versa, so we will often state properties of one that apply equally to the other.

A great deal of ink has been spilled about which signals are in some sense admissable, but we will gloss over most of those issues here. We define the energy in a signal to be

$$E_\infty = \int_{-\infty}^{\infty} |x(t)|^2 dt$$

and note that many signals we work with will have infinite energy. There are mathematical formalisms you can use to handle all of this, but they are beyond the scope of this survey course.

By default we work with complex-valued signals. While real-valued signals are far more common in nature, there are real-valued linear operators whose eigenvalues are not real and thus can complicate intuition and analysis.

1.1 Periodic Signals

We will often deal with periodic signals, where

$$x(t) = x(t + T)$$

The canonical examples being sines and cosines. We will become intimately familiar with the periodic complex exponential signal,

$$x(t) = e^{j\omega t}$$

which, using Euler's formula,

$$e^j\theta = \cos\theta + j\sin\theta$$

can be expressed as

$$e^{j\omega_0 t} = \cos\omega_0 t + j\sin\omega_0 t$$

Periodic signals in the discrete case give rise to our first example of differeng behavior between discrete and continuous time. The complex exponential signal

$$x[n] = e^{j\omega_0 n}$$

has a curious property. Consider the discrete-time complex exponential at frequency $\omega_0 + 2\pi$:

$$x[n] = e^{j(\omega_0+2\pi)n} = e^{j\omega_0 n}e^{j2\pi n} = e^{j\omega_0 n}$$

Note that the signals at frequencies ω_0 and $\omega_0 + 2\pi$ are exactly the same!

Note that also, to achieve periodicity of $x[n]$, we must have

$$e^{j\omega_0 n} = e^{j\omega_0(n+N)}$$

or $e^{j\omega_0 N} = 1$ implying $\omega_0 N$ must be a multiple of 2π, or there must be an integer m such that $\omega_0 N = 2\pi m$. That is, $\frac{\omega_0}{2\pi}$ must be a rational number!

1.2 the unit impulse

The unit impulse, shows up in many areas of electrical engineering, physics, and probability. We will first consider the discrete time version

The discrete-time unit impulse can be used to "sample" a signal at time zero, as

$$x[n]\delta[n] = x[0]\delta[n]$$

The Dirac delta function plays a similar role in the continuous-time domain.

$$\int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0)$$

2 Systems

We speak of systems, which take in a signal $x(t)$ and produce a signal $y(t)$. WE draw them like this. [draw them]

Electrical engineers and computer scientists can bond over our desire to draw boxes and connect them with arrows. It's important to note that a system can look at as much of the signal as it wants, but it also must produce an output for every time.

We could just add stuff together. Or run them in parallel. Or concatenate.

2.1 Memory

A *memoryless* system has no state – the output $y[n]$ can only be a function of the current value of the signal $x[n]$. The system

$$y[n] = x[n]^2$$

is an example of a memoryless system, whereas the system which sums all previous inputs (an accumulator)

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

is not.

2.2 invertability

2.3 causality

A system is causal if the value of $y(t)$ only depends on the values of $x(\tau)$, $\tau < t$. Thus all memoryless systems are causal, but the accumulator described above is also causal.

2.4 stability

A system is stable if bounded inputs produce bounded outputs. The accumulator above is not stable.

2.5 Linearity

This is one of the most important properties a system can have. We're all familiar with linearity so I'll just state that

1. The response to $x_1(t) + x_2(t)$ is $y_1(t) + y_2(t)$
2. The response to $ax(t)$ is $ay(t)$ where $a \in \mathbb{C}$.

This gives rise to superposition! A ton of real-world systems are linear, and will make up the majority of topics in this survey.

2.6 Time Invariance

Time-invariance is another crucial property. Simply stated, a system is time-invariant if a time-shit in the input produces an equivalent time-shift in the output.

If $y(t)$ is the response of a system to an input $x(t)$, the system is time-invariant if, when fed an input $x(t-t_0)$ its output is $y(t-t_0)$.

Many physical systems are time invariant, and even those that aren't can often be modeled locally as time-invariant.

3 Convolution

3.1 Impulse representation of a signal

We can write any input signal as a sum of weighted unit impulse functions. This is easiest to see in the discrete time context. Consider a signal with $x[0] = 3$, $x[1] = 2$, $x[2] = 4$. We can also express this signal as

$$x[n] = 3\delta[n] + 2\delta[n-1] + 4\delta[n-2]$$

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

3.2 Impulse response and convolution

Now, consider a system whose response an impulse at time k, that is, $\delta[n-k]$, is $h_k[n]$. If the system is linear, than we can compute the response of the system to any input signal via

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h_k[n]$$

Note that if the system is also *time invariant*, then $h_k[n] = h[n-k]$, that is the system response to a shifted impulse $\delta[n-k]$ is $h[n-k]$. Then the total system response to an input $x[n]$ will be:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

This is the celebrated convolution operation. We can write this as

$$y[n] = x[n] * h[n]$$

3.3 Continuous-time convolution

Convolution in the continuous case proceeds similarly

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

"Flip and slide"

3.4 Runtime

For discrete-time convolution, convolving a signal $x[n]$ of length N with an impulse response $h[n]$ of length K will take $O(NK)$. This may seem trivial but later we will see how to do much better!

3.5 The linear algebra view

A Toeplitz matrix has constant diagonals

$$G = \begin{bmatrix} h[0] & & & & & & & & \\ h[1] & h[0] & & & & & & & \\ h[2] & h[1] & h[0] & & & & & & \\ & h[2] & h[1] & h[0] & & & & & \\ & & h[2] & h[1] & h[0] & & & & \\ & & & h[2] & h[1] & h[0] & & & \\ & & & & h[2] & h[1] & h[0] & & \\ & & & & & & \ddots & & \\ & & & & & & \ddots & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix}$$

With sufficient padding, the convolution of a signal $x[n]$ with $h[n]$ can be represented as multiplication by a Toeplitz matrix.

Note when we apply this matrix to an impulse, we just get a shifted-and-scaled version of the filter.

3.6 Convolution for hardware

DSPs were important as convolution at each step requires

1. Load a filter coefficient, $h[n]$
2. Load a datapoint, $x[n]$
3. Multiply them
4. Save the result in a register

which is why they had independent data and instruction memories (Harvard architecture), index registers that would auto-increment (zero-overhead indexing), and hardware multiply-accumulate functions in their ALUs.