Signals and Systems II

Part I: Signals and their representations

JEFFREY O. COLEMAN

This six-part series is a minicourse, focused on system concepts, that is aimed at the gap between Signals and Systems and the usual first digital signal processing course.

The goal of this series is a principles-based capability to design signal-processing architectures to convert signals between various forms. Signals are continuous-time or discrete-time, real or complex, and baseband or passband. Operations used on those signals include analog filtering, digital filtering, interpolation, decimation, and frequency shifts. The approach is rigorous but largely graphical with little explicit mathematics.

Design examples focus on systems for sampling, reconstruction, frequency conversion, linear modulation and demodulation, and their various combinations. Among ideas not mentioned are difference equations, z-transforms, discrete-time Fourier transforms, and DFTs. These, the nuts and bolts of DSP, can be studied later, after the student knows what to build with them.

This material has been successfully taught to undergraduates before traditional discrete-time signals and systems. This first article in the series is about motivation, philosophy, and notation.

Systems before DSP? Why?

The fashion has for many years been to consider discrete-time signals and systems and digital signal processing in isolation from the continuous-time world to the extent possible. But is this realistic, or an artifice? Instructors routinely explain algorithms with example signals representing speech, images, radar or communication waveforms, all fundamentally continuous-time, analog quantities. But having seen the DSP world related to the larger continuous-time world only through a quick tour of elementary sampling and reconstruction leaves many engineers a little dizzy whenever they are forced to approach the discrete-/continuous-time boundary and consider how the signals and transforms in one world relate to those in the other. This leads to weaker system design across the profession, as well as to much unnecessary anxiety, and suggests that perhaps it is time to rethink the fundamental philosophical approach to the subject of DSP.

In the beginning, signal processing was about transforming information-bearing signals from available forms to needed forms compatible with various media like circuits, antennas, transmission lines, data links, storage systems, microphones, and speakers through which information must be transmitted, stored, or reproduced. Today this is the elementary aspect of signal processing, and its study focuses on linear systems operating on deterministic signals, which properly should include, in each of continuous time and discrete time, signals that are lowpass, bandpass, real, and complex in basic nature, each compatible with various media. Transformations of such signals cannot be discussed today in meaningful generality with discrete-time signals always considered separately. But which topics in traditional discrete-time signals and systems are actually prerequisite to working with the range of possible simple linear-system structures for changing the forms of signals? There are surprisingly few.

Traditional topics in undergraduate discrete-time signals and systems and DSP include computational convolution, transfer functions, difference equations and associated filter

Digital Object Identifier 10.1109/MPOT.2008.XXXXXX

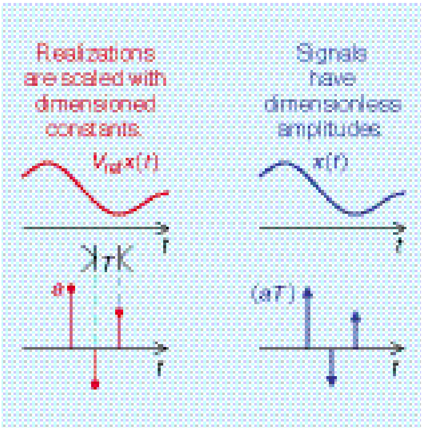

Fig. 1 Signals are mathematical entities, but their realizations, here a voltage and a numerical sequence, are more physical.

structures, and perhaps the plug-n-play design of filters to simple specifications. But all of these are actually about realization or implementation of various processing steps, not about the steps themselves and what they should aim to do. Along the educational way, new transforms enlarge the Fourier menagerie but to a purpose less than readily apparent to many students, who constantly ask, "Why? What can this be used for?" Indeed, without some system context, the study of such implementation matters and the transforms associated with them is unmotivated. Implementing various steps is fine, but how does one decide what steps are appropriate? Where should there be samplers, filters, decimators, and such?

All this suggests a reordering of the conventional topic sequence. Perhaps a general study of linear systems that change deterministic signals from one form to another should *precede* these implementation-related specifics of DSP systems. To do this requires, however, an approach to thinking and talking about discrete-time signals and systems that does not require what we usually, but apparently incorrectly, think of as the major subtopics of the subject. The purpose of this series then is to present such an approach to the conception of systems with embedded DSP, an approach both easily taught to undergraduates who have yet to study discrete-time signals and systems and easily mastered by engineers whose primary specialties are elsewhere. This approach has been used successfully in such contexts several times, primarily with seniors in a core course in nominally analog communications.

The heart of the approach is the separation of the *idea* of a signal from the idea of its *realization* in a piece of hardware or software. Possession of this key idea allows one to keep signals and signal processing locked in a strictly continuous-time world in which "discrete time" takes a different meaning, referring to signals comprising trains of impulses. In this world, there is nothing particularly peculiar or unnatural about hybrid analog/DSP systems, multirate systems, or complex signals. System concepts are worked out in the frequency domain using manipulation of simple Fourier sketches. The only prerequisite to this approach to preliminary system design is familiarity with the basic concepts of continuous-time signals and systems.

When such an approach is used in the classroom, what is the curricular fate of traditional discrete-time signals and systems and DSP? These subjects parallel or follow afterward, repackaged only slightly. The impulse trains that here make up discrete-time signals comprise impulses with areas that, when systematically normalized, become the familiar sample sequences that realize discrete-time signals in computational DSP systems. A traditional DSP course with minor relabeling then becomes the study of computational-DSP systems that operate on such sequences to realize the signal-processing operations explored in this series of articles.

The remainder of this series amounts both to concise teaching notes for an undergraduate presentation by a signals-and-systems instructor and to a self-study mini-text for an engineer with some experience, for a graduate student, or for a motivated undergraduate.

This development is almost entirely graphical, with explicit mathematics required only to establish two simple Fourier-transform properties not typically encountered in treatments of continuous-time signals and systems. The first, immediately below, establishes the nature of discrete-time signals in the frequency domain. The other will come much later, in the discussion of complex signals.

Discrete-time signals

Let us begin by agreeing that a *discrete-time* (d.t.) *signal* $x(t)$ is one that is nonzero only on some discrete and uniformly spaced set of times that, let us say, includes $t = 0$, a condition oddly but conveniently stated for signal $x(t)$ as

$$x(t) = x(t)e^{j2\pi t/T}. \qquad (1)$$

At $t = kT$, an arbitrary integer multiple of T, the exponential becomes $e^{j2\pi kT/T}$ or unity, so $x(t)$ can have any value and the equality will hold, but at other times only $x(t) = 0$ will do. Fourier transforming (1) above to $X(f) = X(f - 1/T)$ shows that $1/T$, the signal's impulse rate, is a period of $X(f)$, *so discrete-time signals are paired with periodic transforms*. Since $X(f)$ is periodic, we can express it using a Fourier series as

$$X(f) = \sum_{n=-\infty}^{\infty} a_{-n} e^{j2\pi n f T},$$

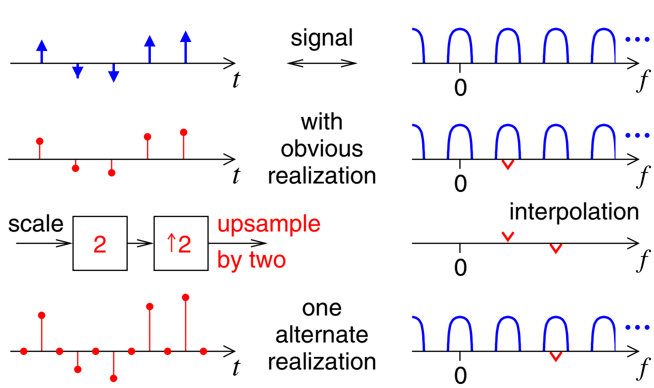

Fig. 2 Interpolation leaves the signal unchanged but transforms its realization to a higher sample rate.

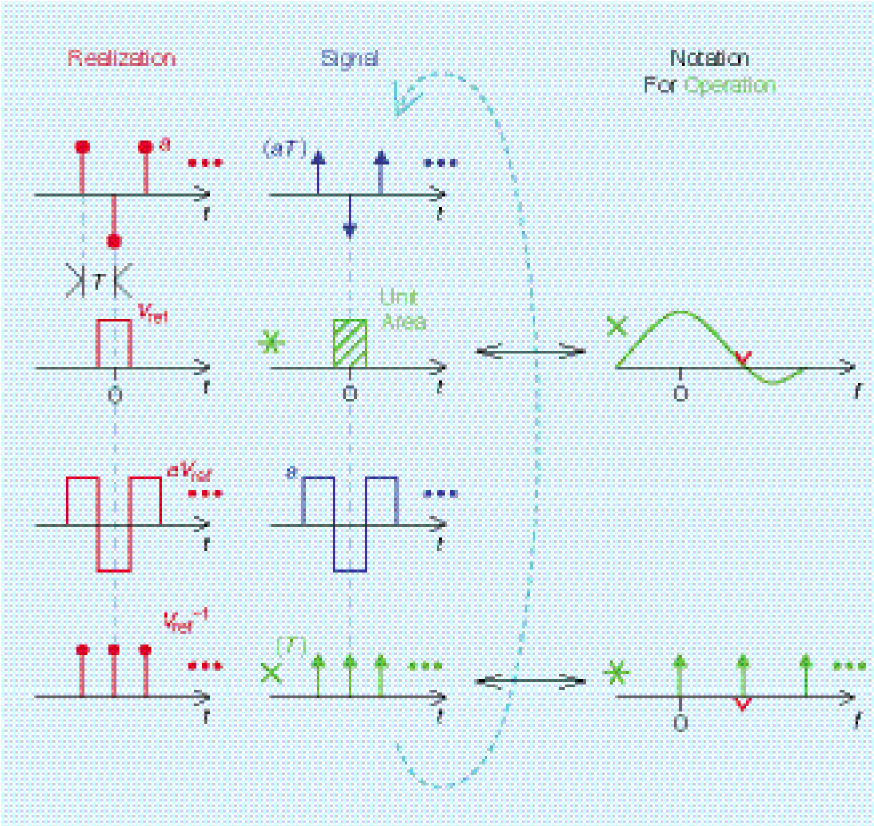

Fig. 3 D/A conversion and sampling, arranged in a loop. From top: discrete-time signal, D/A conversion, conversion output, and sampling (output at top).

where the independent variable is f and the "fundamental" interval is T, with these replacing the familiar t and f_0. The Fourier coefficient for time nT is named a_{-n} to make change of index $n \to -n$ lead conveniently to

$$X(f) = \sum_{n=-\infty}^{\infty} a_n e^{-j2\pi nfT},$$

which in turn has inverse Fourier transform

$$x(t) = \sum_{n=-\infty}^{\infty} a_n \delta(t - nT).$$

From the latter we see that *discrete-time signals are just uniformly spaced impulse trains*. Many will forget these mathematics, the most challenging of this entire series of articles, but all should remember that last sentence!

This series violates two long-established terminological conventions. First, here d.t. signals are functions of a real-valued time variable t but are nonzero only for t of the form nT, where n is an integer. Conventionally however, d.t. signals are written as a function of that integer n, the "discrete" time variable. We will deal further with this issue later.

Second, since we have only the one time variable t and since that time variable is continuous, we can't very well talk about "continuous time" signals as distinct from discrete-time signals as is customary. Instead we will use "analog" in an entirely common but entirely incorrect way to refer to signals that are not d.t. and contain no d.t. component. Of course it is a continuous range rather than a continuous domain that makes a signal $x(t)$ properly termed analog, but this misuse is so widespread as to be a *de facto* standard, and every electrical engineer needs to be familiar with both usages.

Signals versus realizations

For us, *signals* are just functions of time generalized to permit impulses, hence they are abstractions existing in our minds, on paper, or in our analyses. In implementations of systems these signals have various representations or *realizations*. Analog signals are typically realized as voltages or currents, but any physical quantity controllable over a continuum of values will do. Our signal amplitudes are always dimensionless, but dimensionless signal $x(t)$ might be realized, for example, as voltage $V_{\text{ref}}\,x(t)$ as in Fig. 1, with reference voltage V_{ref} chosen for implementation convenience. Its choice is a hardware-design issue.

Because signal amplitudes are dimensionless, signal impulse areas have time dimensions and an impulse area is naturally realized in normalized form as dimensionless number area/T_{ref}, that is, area aT_{ref} is naturally realized as dimensionless number a. A discrete-time signal is then realized computationally in hardware or software as a *sequence* of such numbers, called *samples*, occurring at some *sample rate* $1/T$. For the uniformly spaced impulses of a d.t. signal we can simply choose constant T_{ref} as the sample spacing T, as in Fig. 1.

Is the sample rate of the realization unique? Does the upper-left signal of Fig. 2 comprise simply the widely spaced impulses that are visible? Or does it actually consist of narrowly spaced impulses with many of their areas zero? The second and fourth lines of the figure show two of many possible realizations as sequences. Notice that halving the standard T_{ref}, sample spacing T, doubles the scale of the samples by changing their normalization. The ambiguity is resolved below by adopting a standard notation for a d.t. signal that explicitly indicates the sample rate to be used in its realization.

All d.t. signals were assumed above to be realized computationally, making

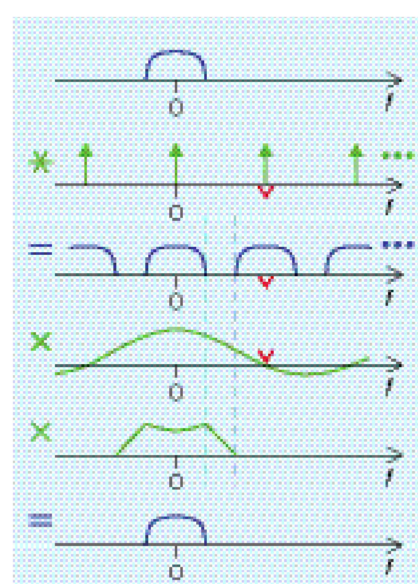

Fig. 4 Sampling and reconstruction of the signal sampled.

the processing of d.t. signals equivalent to *digital signal processing* or DSP. But one might just as well scale dimensionless signal amplitude by some reference current I_{ref} to make a signal impulse of area aT correspond to a current impulse with area equal to some quantity $aI_{ref}T$ of electric charge, with I_{ref} chosen for implementation convenience. Such charge packets can be manipulated electronically with *switched-capacitor systems*, and most of the development to follow applies to systems of that type as well as to DSP-based systems.

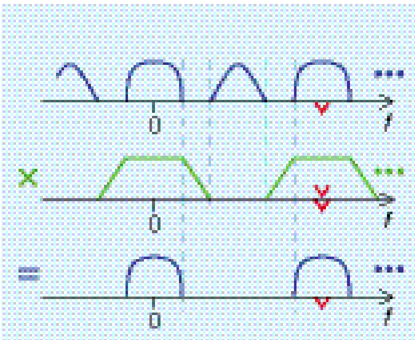

Fig. 5 A digital filter applies a periodic frequency response to a discrete-time input.

Notation for signals

Our standard signal notation is a schematic sketch in the frequency domain that mirrors the properties of a signal, showing it perhaps to be impulsive or bandlimited or symmetric. The example d.t. signal on the top line of Fig. 2 is shown in both time and frequency domains. In the spectral sketch, ellipses on the right indicate the spectral periodicity that flags a signal as d.t.

The only feature of the notation here that is likely to seem unfamiliar to most DSP engineers is the triangular tick mark, which indicates the sample rate to be used in the corresponding realization. This sample-rate tick has nothing to do with the signal itself and so can be omitted in the rare cases in which the realization is not of interest. The period sketched for a signal's transform implies its impulse rate, and most often it will match the tick-marked sample rate. Note that realizing impulse area aT as dimensionless sample a is equivalent to scaling that area by the tick-marked sample rate $1/T$ in moving from the signal to its realization. In this series the "scaling" and the "normalization" of the samples are referred to more or less interchangeably.

Early DSP texts assumed that a single sampling rate was used in realizing all d.t. signals in a system, but here no such assumption is made, because in modern systems *multirate* systems are the norm. For the same reason, frequencies will never be normalized to the sampling rate but will always be actual and honest physical frequencies.

D/A conversion

For the present purposes of enabling a basic level of abstract system design, *digital-to-analog* (D/A) *conversion* refers to filtering a d.t. input with any impulse response that is not d.t. The top three quarters of Fig. 3 shows such a conversion. A standard conversion uses an impulse response, as shown here, that is a unit-area rectangle centered about the time origin with width equal to the sample interval. Unit area implies unity dc gain and so makes the associated $\mathrm{sinc}(fT)$ frequency response easy to sketch. Even if we did not recognize filter's frequency response as a sinc, we could still see that input sinusoids that fit only complete cycles into impulse-response width T must integrate to zero in the convolution and therefore that this frequency response must have nulls at the sample rate and its nonzero multiples.

The noncausality of the rectangular impulse response shown of course makes it unrealizable, but that is of no more interest or significance than the propagation delays generally omitted from idealized models of other circuits and computational systems. Just as we can model propagation delay explicitly when its effects are important, here we can explicitly model an additional half-sample delay on those rare occasions when its effects would be of interest. We will treat other filters similarly, advancing time with noncausal filters when doing so harmlessly simplifies the mathematics.

Notation for operations

In our notation for signal-processing operations that act on signals, the input is to be combined in the frequency domain with the function sketched, using the operator at its left. Triangular ticks above and below the axis mark the input and output realization rates (and normalizations) respectively. The sinc sketch in Fig. 3 is therefore flagged as a D/A conversion by the presence of only an input tick. There is no output tick because the output is not a d.t. signal. A thorough notation would also indicate the reference voltage or current by which the realization output is scaled, but here we simplify and consistently omit the dimension-carrying realization scale constants of analog signals. In the Fig. 3 case, the time and voltage reference levels could be absorbed into a realization unit-sample response with amplitude equal to the voltage reference.

Sampling

Multiplying d.t. signals would of course be as nonsensical as multiplying two co-located impulses anywhere but in convolution-like integrals. But two waveforms are easily multiplied when one is d.t. and the other is continuous at the impulse times of the first. Such products appear routinely in three contexts in DSP. The first is the *sampling* operation, an example of which appears in the bottom line of Fig. 3. The others, decimation and sinusoidal modulation, are discussed later.

In Fig. 3 an input signal, the third-line square wave, is multiplied by a fourth-line sampling waveform that comprises impulses at some rate $1/T$ and of uniform area T. In the frequency domain this is convolution "$*$" with the Fourier transform of the sampling waveform, which consists of unit-area spectral impulses at multiples of the impulse rate.

The latter transform is easily derived using two Fourier properties: first, that periodicity in either domain corresponds to uniformly spaced impulses in the other and, second, that the time-domain average equals the area of the dc impulse.

It follows that the sampling operation's output spectrum comprises periodically shifted copies of the input spectrum. Of course physical A/D converters also quantize signals and therefore introduce approximation effects, but these are beyond the scope of this discussion.

The output tick here formally denotes the rate and normalization of the output samples of the analog-to-digital (A/D) converter that realizes the sampling operation. That tick can also be reasonably viewed as applying to the sampling waveform viewed as a signal.

Signal reconstruction

The first two lines of Fig. 3 show a signal-reconstruction operation, realized as D/A conversion, that "undoes" the sampling just described. The square-wave signal shown happens to make the sampling and reconstruction operations into exact inverses. While sampling and reconstruction is often depicted for restricted classes of signals, this square-wave restriction is obviously too severe to be useful.

A more typical example appears in Fig. 4 using, as is quite common, a frequency-domain description only. There a bandlimited real lowpass signal, perhaps music in a recording studio, is converted to d.t. form, perhaps for storage on or transmission through some digital medium, and back again. The system is described by a spectral sketch that states algebraic relationships in the frequency domain. Read the lines from top to bottom: *first* ∗ *second* = *third* and (*third* × *fourth*) × *fifth* = *sixth*.

Upsampling and interpolation

The third line of Fig. 2 represents a processing step that increases the realization sample rate by some integer factor without affecting the signal itself. Normalization in the realization is also affected, with the net effect being simply to scale up the incoming samples by the tick-frequency ratio. Inserting zero samples into the sample sequence *in the realization* to increase the sample rate by an integer factor M is *upsampling by* M, denoted $\uparrow M$. This series of articles strains terminological convention by using *interpolation* to refer to the tick-mark-shifting null signal operation on the right, which operation corresponds both to upsampling and renormalization scaling in the realization.

Digital filtering

A *digital filter* has a d.t. impulse response and therefore a periodic frequency response, and it operates on a d.t. input signal. In this series the term is restricted to the common case depicted in Fig. 5 in which input and output sample rates are identical and both equal to the frequency-response period. This makes computing the convolution in the realization relatively straightforward. In practice, however, digital filtering in this sense and a preceding or following operation are often realized jointly for the sake of efficient implementation.

Part II will continue with discussions of oversampling in D/A conversion and the basics of decimation, complex signals, and Nyquist signaling.

Read more about it

The material in the first three chapters of the textbook below is typical of coverage of d.t. matters in an introductory course in signals and systems. The remainder of that text's material would typically be covered in a classic DSP course at the advanced undergraduate or first-year graduate level.

A.V. Oppenheim and R. Shafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1993.

Figures are posted for instructional use on the author's Web site, available at http://alum.mit.edu/www/jeffc.

About the author

Jeffrey O. Coleman (S'75-M'79-SM'99) joined the Radar Division of the Naval Research Laboratory (NRL) in Washington, D.C. in 1978. He left NRL in 1985 for graduate studies, a stint with The Boeing Company, and a faculty position at Michigan Technological University. He returned to NRL in 1997.

His 1975/1979/1991 SBEE/MSEE/Ph.D. degrees are from the Massachusetts Institute of Technology, Johns Hopkins University, and the University of Washington respectively, and his research is on theory and design methods in DSP.