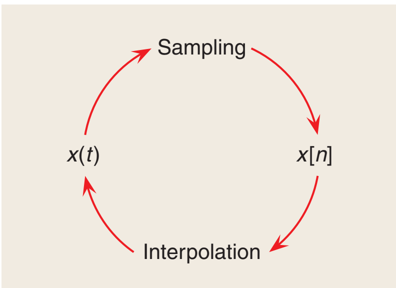
Paolo Prandoni and
Martin Vetterli

From Lagrange to Shannon ... and Back: Another Look at Sampling

Classical digital signal processing (DSP) lore tells us the tale of a continuous-time primeval signal, of its brutal sampling, and of the magic sinc interpolation that, under the aegis of bandlimitedness, brings the original signal back to (continuous) life. In this article, we would like to switch the conventional viewpoint and cast discrete-time sequences in the lead role, with continuous-time signals entering the scene as a derived version of their gap-toothed archetypes. To this end, we will bring together some well-known but seldom-taught facts about interpolation and vector spaces and recall how the classic sinc reconstruction formula derives naturally from the Lagrange interpolation method. Such an elegant and mathematically simple result can have a great educational value in building a solid yet very intuitive grasp of the interplay between analog and digital signals.

Traditionally, the standard pedagogical approach to discrete-time signal processing has long suffered from what we may call a "platonic complex" of sorts: much as in the myth of the cave, the discrete-time world is often introduced as a tributary notion to the "real" continuous-time universe. If we consider the merry-go-round in Figure 1, which links discrete- and continuous-time signal processing, it is clear that one can choose either domain as a starting point. Historically, however, when electronic circuits and analog design still constituted the backbone of applied signal processing and communications, teaching practice would invariably start with continuous time and only later introduce the "magic" sampling and interpolation formula credited to Shannon [1]. Most of the classic textbooks used today still follow this approach [2], [3].

On the other hand, the majority of our information sources now are digital, and most of the data that we enjoy in myriad different formats has basically been "born digital." With the Internet seeming to displace physical reality, it is tempting to introduce signal processing by way of discrete-time sequences first, and indeed recent books have embraced a view in which signals are considered primarily as computer-generated sequences [4], [5]. By building on standard linear algebra, one can quickly assemble a workable set of signal processing notions, a task greatly aided by the ready availability of numerical packages and online interactive applications. A difficulty in this otherwise compelling hands-on approach is the pesky transition from the discrete to the analog world, which sooner or later must be tackled to reconcile our physical senses with digital entities. If the focus is mainly on the operational side of things (by insisting, for example, on practical interpolation circuits), one fails to build a solid bridge between the two different abstractions represented by discrete and continuous time. More often than not, sampling and interpolation end up retaining an esoteric sheen.



[FIG1] The time domain merry-go-round (aliasing not included).

Starting to explain signal processing from the discrete-time point of view is, however, an undeniably "natural" approach: in practice, any observational experience involving real-world signals amounts to a countable series of measurements (think of your primary school projects involving precipitation levels or seasonal temperature records). Luckily, the transition to the continuous-time paradigm can be mathematically meaningful without sacrificing immediateness, and it is surprising that the way to do so does not appear more prominently in the signal processing curriculum. As we will illustrate shortly, a simple algebraic technique called Lagrange interpolation leads naturally to the sinc reconstruction formula. Polynomial interpolation schemes such as Lagrange's emerged centuries ago in the context of practical, hands-on numerical problems so that their connection to the sampling theorem has a great pedagogical value, together with the added virtue of requiring only elementary mathematics. Similarly, the same linear algebra framework can be used to complete the circle in Figure 1 by casting Shannon sampling as a basis expansion in a Hilbert space. In so doing, bandlimitedness appears as a structural property of the space rather than a somewhat mysterious (if physically meaningful) requirement of the analog signal.

Today, many promising research topics in signal processing are spurred by a keen interest in nonsubspace sampling. Compressed sensing, compressive sampling, and sparse sampling, for instance, are all extensions of the Shannon sampling framework in which the classic subspace structure is no longer applicable [6]. We like to think that, amidst the excitement of many new theorems, "revisiting the classics," as we are doing

Digital Object Identifier 10.1109/MSP.2009.933381

 1053-5888/09/$26.00©2009IEEE

here, and finding long threads of historical continuity is not only a nice exercise but also a source of inspiration.

FROM DISCRETE TO CONTINUOUS TIME

Consider an N-point discrete-time signal $x[n]$, and look at the problem of manufacturing an associated continuous-time version $x(t)$. The first step is to move from the abstract integer index n to a veritable time interval; to do so, we must define our time base, which is simply how much time we "wait" between sample values. Call this duration T, and note that T is now expressed in actual seconds. Then, we need to somehow draw a function of time $x(t)$ that "fills the gaps" between the beginning $(t = 0)$ and the end $(t = (N-1)T)$ of this brand-new continuous-time signal. While there is clearly an infinite number of ways to do that, the following two requirements seem to be all but unavoidable:

- the two signals' values must coincide where discrete and continuous times meet: $x(nT) = x[n]$
- the continuous-time signal should be smooth.

Smoothness is a very natural requirement for a signal that is supposed to model a real-world phenomenon. Just as Nature abhors a vacuum, she likewise abhors abrupt jumps, either in magnitude or in slope. Happily, ultimate smoothness is very simple to express mathematically since it corresponds to the existence of an infinite number of continuous derivatives; in other words, we require $x(t)$ to be of class C^∞, where the notation C^N indicates the set of functions possessing N continuous derivatives. Conveniently, the maximally differentiable curve through a set of N data points is known to be the unique polynomial interpolator of degree $N-1$

$$P(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_{N-1} t^{(N-1)} \qquad (1)$$

which, like all polynomials, belongs to C^∞. Computation of the interpolator's N coefficients is a classic algebraic problem, thoroughly solved in the 17th century by Newton, among many others. Numerically, one way to arrive at the solution is to work out the system of N equations

$$\{P(nT) = x[n]\}_{n=0,1,\ldots,N-1},$$

which can be carried out algorithmically using Pascal's triangle, for instance. However, a much more interesting approach is to consider the vector space of finite-degree polynomials over an interval: any element of such a space can be expressed as a linear combination of simple polynomial basis vectors. The interpolator in (1) is clearly written out as a linear combination of the monomial basis vectors $\{1, t, t^2, \ldots, t^{N-1}\}$, but for the task at hand a much more interesting and appropriate basis is the set of Lagrange polynomials.

Before we explore the concept in more detail, let us make a couple of notational simplifications, which entail no loss of generality. First, let's set $T = 1$ unit of time, so that we can get rid of one unnecessary symbol. Second, let's consider finite-length signals whose support is symmetrical around $n = 0$, i.e., odd-length signals of length $2N+1$ extending from $n = -N$ to $n = N$; with $T = 1$ the continuous-time interpolation interval is therefore the closed interval $I = [-N, N]$ and the interpolation nodes are the integers from $-N$ to N. The uniform Lagrange polynomial basis for the interval I is the family of $2N+1$ polynomials

$$L_n^{(N)}(t) = \prod_{k=-N,\,k\neq n}^{N} \frac{t-k}{n-k}, \qquad n = -N, \ldots, N, \qquad (2)$$

each of which is a polynomial of degree $2N$. As an example, the family of five polynomials for $N = 2$ is shown in Figure 2. If we use Lagrange polynomials, writing out the smoothest interpolator for a discrete-time signal of length $2N+1$ becomes a borderline trivial task; indeed we have simply

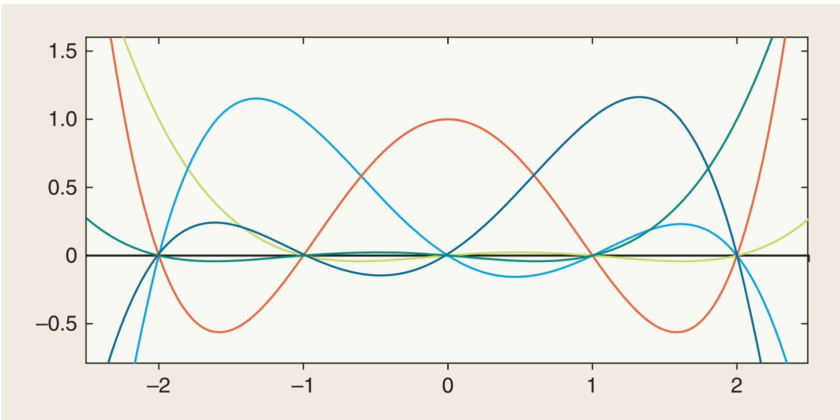
$$x_L(t) = \sum_{n=-N}^{N} x[n] L_n^{(N)}(t) \qquad (3)$$

or, in words, we have that the smoothest interpolator is simply the linear combination of the Lagrangian basis vectors in which the scalar coefficients are the discrete-time samples themselves. The validity of the above statement comes from two facts: 1) $x_L(t)$ above is indeed a polynomial of degree $2N$, and 2) $x_L(t)$ indeed interpolates the discrete-time signal since an immediately verifiable property of Lagrange polynomials is

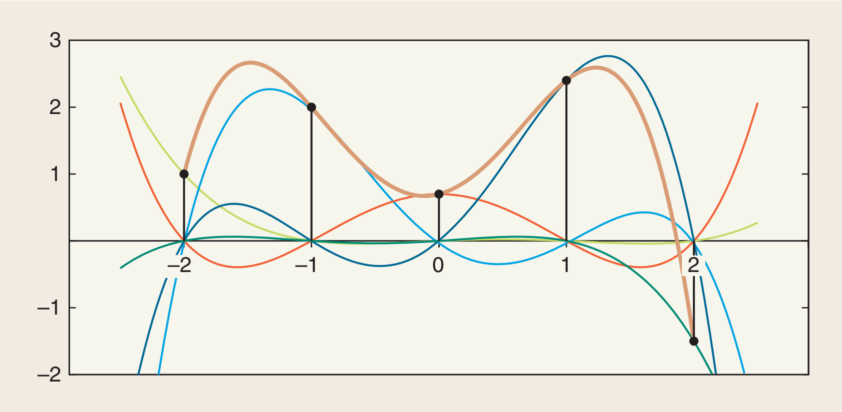
$$L_n^{(N)}(m) = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \qquad -N \le n, m \le N$$

from which $x_L(n) = x[n]$. As an example, Figure 3 shows how the five polynomials $L_n^{(2)}(t)$ beautifully come together to interpolate a five-tap signal; note that, at all times, all five basis polynomials contribute to the instantaneous value of the interpolated signal.

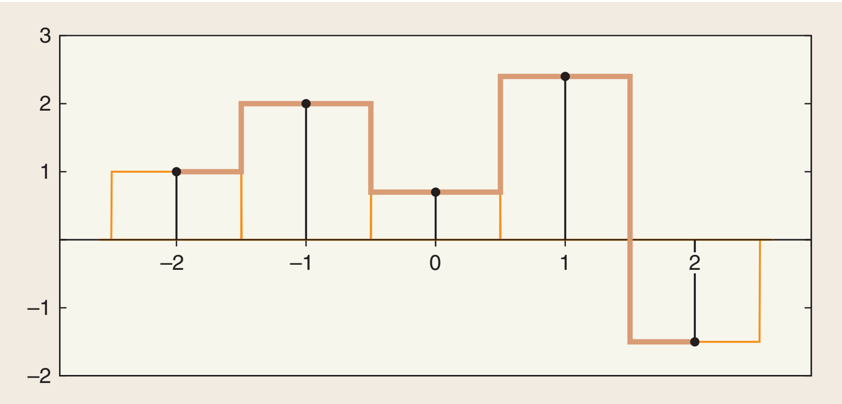
Although it elegantly solves the problem at hand, Lagrangian interpolation suffers from a couple of very serious drawbacks, the worst of which is that the set of basis functions changes completely for



[FIG2] The family of Lagrange polynomials $L_n^{(2)}(t)$; $L_0^{(2)}(t)$ is drawn in red, $L_{\pm1}^{(2)}(t)$ in blue and $L_{\pm2}^{(2)}(t)$ in green, with lighter shades indicating negative indices.



[FIG3] **Maximally smooth Lagrange interpolation of a length-five signal.**



[FIG4] **ZOH interpolation of a length-five signal; shown in orange are the delayed and scaled replicas of the rectangular interpolating kernel $I_0(t)$.**

different signal lengths. From an engineering point of view, it is obvious that we would like a more universal interpolation machine rather than having to redesign our interpolator as a function of the input. The second drawback, which is just as bad, is that polynomial fitting tends to become numerically unstable when the polynomial degree grows large. A way out of the impasse is to relax at least partially the smoothness constraint. This leads to simpler interpolation schemes that, very attractively from a practical point of view, do not suffer from either limitation. Consider for instance the classic zero-order hold (ZOH): all the rest being equal (interpolation interval and so on) the ZOH operates by producing a continuous-time signal $x_0(t)$ in which the signal's value is kept constant around interpolation nodes

$$x_0(t) = x[\lfloor t + 0.5 \rfloor], \quad -N \le t \le N,$$

which is shown in Figure 4 for the same five-point signal as in Figure 3. The above expression can be rewritten as

$$x_0(t) = \sum_{n=-N}^{N} x[n]\mathrm{rect}\,(t-n), \qquad (4)$$

which highlights several interesting facts. First of all, (4) looks a lot like (3), except that now the continuous-time term in the above sum is no longer dependent on either the length of the interpolation or on the discrete-time index in ways other than a simple time shift. The ZOH therefore creates a continuous-time signal by stitching together delayed and scaled versions of the same prototype function, thereby effectively implementing an interpolation machine that works independently of the signal's length. Conceptually (and electronically), this defines a very simple device in which the instantaneous output value depends only on the most recent discrete-time value injected in the interpolator; the interpolation scheme is therefore local, in the sense that it uses only a fixed (and finite) number of input values at a time. By contrast, the Lagrange interpolator is global, since all discrete-time points are necessary to produce any single value of the continuous-time signal.

We can generalize the ZOH's paradigm to encompass local interpolators for which the output has the form of the "mixed-domain" convolution

$$x(t) = \sum_{n=-N}^{N} x[n]I(t-n), \qquad (5)$$

where $I(t)$, called the kernel, is a compact-support function fulfilling the fundamental interpolation properties

$$\begin{cases} I(0) = 1 \\ I(t) = 0 & \text{for } t \text{ a nonzero integer.} \end{cases}$$

Note that the kernel is time invariant in that it does not depend on the time index; the combination of this property with the linearity of the formula in (5) shows that we can interpret interpolation as a "filtering" operation across time domains. Let us consider the first-order kernel, for instance

$$I_1(t) = \begin{cases} 1 - |t| & \text{if } |t| < 1 \\ 0 & \text{otherwise;} \end{cases}$$

the triangular function $I_1(t)$, which is of class C^1, provides an interpolated signal that is the basic "connect-the-dots" line between the discrete-time points, as shown

$$I_3(t) = \begin{cases} 1.25|t|^3 - 2.25|t|^2 + 1 & \text{for } |t| \le 1 \\ -0.75(|t|^3 - 5|t|^2 + 8|t| - 4) & \text{for } 1 < |t| \le 2 \\ 0 & \text{otherwise.} \end{cases}$$

in Figure 5. Again, this is a local interpolation machine since all values of the continuous-time signal depend on at most two neighboring discrete-time samples. Next in line is the cubic interpolator [7] (even-degree kernels are not considered since they are not centered in zero), shown in the box at the bottom of the previous page. This kernel's support is four, and, as a consequence, the values of the continuous time signal depend at most on four neighboring discrete-time samples; furthermore, the kernel is designed to be of class C^2. The cubic interpolation of the by-now-familiar five-tap signal is shown in Figure 6.

The main drawback of local interpolation schemes lies in the limited smoothness of their output. This is especially apparent in the ZOH, where the result is blatantly discontinuous; the situation starts to improve in the first-order interpolator, for which the output is continuous but the first derivative is not, and improves even further with the third-order kernel, which provides a twice-differentiable output. In general, every time we increase the kernel's support, we can gain extra degrees of continuity but, ultimately, the last derivative will be discontinuous; similar results can be obtained with local interpolation schemes based on B-splines of increasing order [1], [8].

And so, in the end, the conundrum seems to be either we choose a smooth but complex and instability-prone interpolation scheme, or we choose a simple and stable scheme but whose smoothness leaves something to be desired. In fact, a little miracle is in store: if we go back to the maximally smooth but nonlocal Lagrangian interpolation, it turns out that, as we increase the width of the interpolation interval to infinity, all the polynomials in the associated basis end up converging to shifted replicas of the very same prototype function! This function, by now an easy guess, is none other than the sinc, viz:

$$\lim_{N\to\infty} L_n^{(N)}(t) = \mathrm{sinc}\,(t-n), \qquad (6)$$

and so the ideal interpolator, in the sense of an interpolator that is both kernel-based and maximally smooth, happens to be the sinc, yielding the well-known

$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{sinc}\,(t-n). \qquad (7)$$

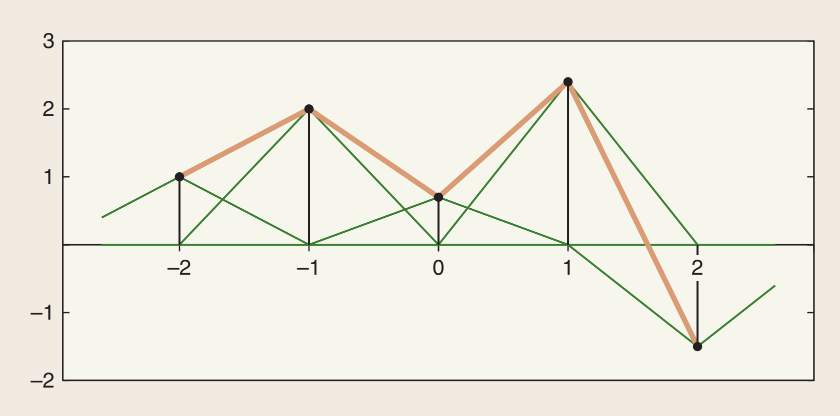
The interpolator is ideal in a "platonic" way as well now, since it has infinite support and can only be aspired to but never arrived at; and so the sinc has brought us back to the classical path to continuous time. Before we proceed to give a proof of this remarkable result, please note that at no point in our discussion did the spectral properties of the sinc function come into play; the sinc just emerges as the ideal interpolation kernel bridging the naturally smooth polynomial interpolation with the algorithmic approach of time-invariant interpolation schemes.

As for a proof of (6) we are partial to Euler's 1748 attempt that, while lacking in rigor by modern standards, has all the irresistible charm of derivations relying only on basic calculus [9]. The argument begins by considering the N roots of unity for N odd. They will be $z = 1$, plus $N-1$ complex conjugate roots of the form $z = e^{\pm j\omega_N k}$ for $k = 1, \ldots, (N-1)/2$ and $\omega_N = 2\pi/N$. If we group the complex conjugate roots pairwise we can factor the polynomial $z^N - 1$ as

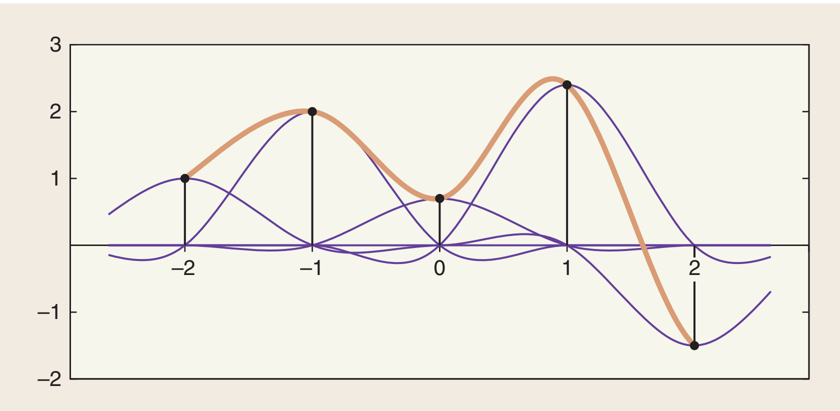
$$z^N - 1 = (z-1)\prod_{k=1}^{(N-1)/2}(z^2 - 2z\cos(\omega_N k) + 1)$$

and the above can be quickly generalized to

$$z^N - a^N = (z-a)\prod_{k=1}^{(N-1)/2}(z^2 - 2az\cos(\omega_N k) + a^2).$$



[FIG5] First-order interpolation of a length-five signal; shown in green are the delayed and scaled replicas of the triangular interpolating kernel $I_1(t)$.



[FIG6] Third-order interpolation of a length-five signal; shown in purple are the delayed and scaled replicas of the cubic interpolating kernel $I_3(t)$.

Now replace z and a in the above formula by $z = (1 + x/N)$ and $a = (1 - x/N)$, as in the box at the bottom of the page, where A is just the finite product $(4/N)\prod_{k=1}^{(N-1)/2}(1-\cos(\omega_N k))$. The value A is also the coefficient of the first degree term in the right-hand side and it can be easily seen from the expansion of the left hand-side that $A = 2$ for all N; actually, this is an application of Pascal's triangle and it was proven by Pascal in the general case in 1654. Now, let the lack of rigor begin: as N grows large we know that

$$\left(1 \pm \frac{x}{N}\right)^N \approx e^{\pm x},$$

at the same time, if N is large, then $\omega_N = 2\pi/N$ is small and, for small values of the angle, the cosine can be approximated as

$$\cos(\omega) \approx 1 - \omega^2/2$$

so that the denominator in the general product term can in turn be approximated as

$$N^2(1-\cos(2k\pi/N)) \approx N^2 \cdot \frac{4k^2\pi^2}{2N^2} = 2k^2\pi^2.$$

Finally, for large N, the numerator can be approximated as $1 + \cos(2k\pi/Nk) \approx 2$ and therefore the above expansion becomes (by bringing $A = 2$ over to the left-hand side)

$$\frac{e^x - e^{-x}}{2} = x\left(1+\frac{x^2}{\pi^2}\right) \times \left(1+\frac{x^2}{4\pi^2}\right)\left(1+\frac{x^2}{9\pi^2}\right)\cdots.$$

The last trick is to replace x by $j\pi t$ and voila

$$\frac{\sin(\pi t)}{\pi t} = \prod_{n=1}^{\infty}\left(1 - \frac{t^2}{n^2}\right).$$

A more rigorous proof can be found in "The Sinc Product Expansion Formula."

DSP, LINEAR ALGEBRA, AND HILBERT SPACES

While describing Lagrangian interpolation, we have already exploited the convenience of vector spaces to quickly identify the role of the main ingredients in the interpolation recipe. We will now carry this one step further and introduce a precise mathematical framework in which signals are equivalent to elements of a suitable vector space; this will help us immensely for our next task, which is closing the loop from continuous time back to discrete time.

The initial step is to consider the world of finite-length discrete-time signals, where all is always well; this is the eminently practical domain of DSP algorithms, it is the actionable world of numerical packages such as MATLAB, it is, in short, the world of standard linear algebra [10]. The equivalence between an N-point finite-length discrete-time signal and a vector in Euclidean $\mathbb{C}^N$ is immediate to see as the N-point signal $x[n]$, with $0 \le n < N$, can be expressed as the (column) vector

$$\mathbf{x} = [x[0]\ x[1]\ \ldots\ x[N-1]]^T.$$

Another useful way to look at the above equivalence is to consider the canonical basis for $\mathbb{C}^N$ $\{\mathbf{e}^{(k)}\}_{k=1,\ldots,N-1,}$ where $\mathbf{e}^{(k)} \equiv \delta[n-k]$; as for any given basis, we can express any vector in $\mathbb{C}^N$ as a linear combination of basis vectors and in this case we have the straightforward

$$\mathbf{x} = \sum_{k=0}^{N-1} x[k]\mathbf{e}^{(k)}.$$

Once the signal-vector analogy is established, we can rely on geometric intuition to easily introduce and explain the whole signal processing machinery. We know for instance that the standard inner product in $\mathbb{C}^N$ is a measure of similarity between vectors

$$\langle \mathbf{x}, \mathbf{y}\rangle \equiv \langle x[n], y[n]\rangle = \sum_{i=0}^{N-1} x^*[i]\,y[i].$$

As a consequence, the convolution operation (which is just an inner product with a circular shift) is easily understood as a localized measure of similarity between a candidate vector (the input signal) and a prototype vector (typically, an impulse response). From the point of view of linear algebra, the discrete Fourier transform is just a change of basis in $\mathbb{C}^N$ in which we replace the canonical basis $\{\mathbf{e}^{(k)}\}$ with the orthonormal Fourier basis $\{\mathbf{w}^{(k)}\}$, where $\mathbf{w}^{(k)} \equiv w_k[n] = (1/\sqrt{N})e^{j2\pi nk/N}$. Again, this is conveniently expressed as a matrix-vector multiplication

$$\mathbf{X} = \mathbf{W}\mathbf{x},$$

where the elements of the matrix are $\mathbf{W}_{nk} = (1/\sqrt{N})e^{-j(2\pi/N)nk}$. Consequently, each new coordinate in the Fourier basis is the inner product of the "old" vector with one of the Fourier basis vectors, i.e., each new coordinate is a measure of similarity between the old vector and a sinusoid at a given frequency; Fourier analysis, therefore, "turns the space around" to discover and highlight hidden signal properties. Even more explicitly, we can write out the generic analysis and reconstruction formula

$$\mathbf{x} = \sum_{k=0}^{N-1} \langle \mathbf{w}^{(k)}, \mathbf{x}\rangle\, \mathbf{w}^{(k)}, \tag{8}$$

which shows how any vector in $\mathbb{C}^N$ can be expressed as a linear combination of sinusoids where the weighting terms are none other than the Fourier coefficients. Note that this formula is valid for any orthonormal basis and we will use it to much effect in the next section.

The power of the abstraction provided by vector spaces is twofold: first of

$$\begin{aligned}
\left(1+\frac{x}{N}\right)^N - \left(1-\frac{x}{N}\right)^N &= \frac{4x}{N}\prod_{k=1}^{(N-1)/2}\left(1-\cos(\omega_N k) + \frac{x^2}{N^2}(1+\cos(\omega_N k))\right) \\
&= \frac{4x}{N}\prod_{k=1}^{(N-1)/2}(1-\cos(\omega_N k))\left(1 + \frac{x^2}{N^2}\cdot\frac{1+\cos(\omega_N k)}{1-\cos(\omega_N k)}\right) \\
&= Ax\prod_{k=1}^{(N-1)/2}\left(1 + \frac{x^2\,(1+\cos(\omega_N k))}{N^2\,(1-\cos(\omega_N k))}\right)
\end{aligned}$$

all, as in the case of finite-length signals, it provides an intuitive and consistent explanation of discrete-time signal processing that is steeped in familiar geometrical concepts. But, more importantly, it also represents a paradigm that can be extended to entities that require more mathematical subtlety than the good old tangible finite-length sequences, namely infinite-length signals and continuous-time signals. The mathematical framework that encompasses these many worlds is the Hilbert space [11]–[13], defined in abstract terms as a vector space possessing an inner product operator and the property of completeness. The basic operations in a vector space are rather intuitive; for the space of infinite-length discrete-time sequences, for instance, the inner product is defined as

$$\langle \mathrm{x}, \mathrm{y} \rangle = \sum_{i=-\infty}^{\infty} x^*[i]\, y[i],$$

(which of course raises all sorts of convergence questions; for the purpose of this article, let's remain on the conservatively safe side by restricting ourselves to the space of absolutely summable sequences). The convolution operator is easily derived as

$$(x * y)[n] \equiv \langle x^*[m], y[n-m] \rangle.$$

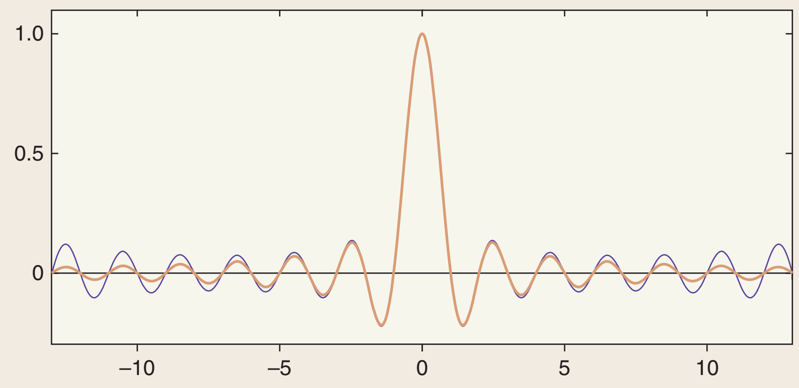
For continuous-time signals (that is, functions), the inner product is defined as

$$\langle \mathrm{x}, \mathrm{y} \rangle = \int_{-\infty}^{\infty} x^*(t) y(t) dt$$

so that

$$(x * y)(t) \equiv \langle x^*(\tau), y(t-\tau) \rangle.$$

Completeness, on the other hand, is a rather technical concept ensuring that the results of limiting operations remain within the vector space (i.e., it guarantees that the space is not, say, like the set of rational numbers where there are sequences converging to irrational limits). Again, for the scope of this article, completeness will be tacitly assumed.



[FIG7] A portion of the sinc function and its Lagrange approximation $L_0^{(100)}(t)$ (thin line).

BACK TO DISCRETE TIME

The search for the ideal interpolation scheme has forced us to abandon the comforts of finite-dimensional vector spaces and tread the subtler grounds of infinite time spans. As this point, we may

THE SINC PRODUCT EXPANSION FORMULA

Consider the Fourier series expansion of the even function $f(x) = \cos(\tau x)$ over $[-\pi, \pi]$ and made periodic over $\mathbb{R}$. We have

$$f(x) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(nx)$$

with

$$\begin{aligned} a_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} \cos(\tau x) \cos(nx) dx \\ &= \frac{2}{\pi} \int_0^{\pi} \frac{1}{2} [\cos((\tau+n)x) + \cos((\tau-n)x)] dx \\ &= \frac{1}{\pi} \left[\frac{\sin((\tau+n)\pi)}{\tau+n} + \frac{\sin((\tau-n)\pi)}{\tau-n} \right] \\ &= \frac{2\sin(\tau\pi)}{\pi} \frac{(-1)^n \tau}{\tau^2 - n^2} \end{aligned}$$

so that

$$\cos(\tau x) = \frac{2\tau \sin(\tau\pi)}{\pi} \times \left(\frac{1}{2\tau^2} - \frac{\cos(x)}{\tau^2 - 1} + \frac{\cos(2x)}{\tau^2 - 2^2} - \frac{\cos(3x)}{\tau^2 - 3^2} + \cdots \right)$$

In particular, for $x = \pi$ we have

$$\cot(\pi\tau) = \frac{2\tau}{\pi} \left(\frac{1}{2\tau^2} + \frac{1}{\tau^2 - 1} + \frac{1}{\tau^2 - 2^2} + \frac{1}{\tau^2 - 3^2} + \cdots \right)$$

which we can rewrite as

$$\pi \left(\cot(\pi\tau) - \frac{1}{\pi\tau} \right) = \sum_{n=1}^{\infty} \frac{-2\tau}{n^2 - \tau^2}$$

If we now integrate between zero and t, both sides of the equation we have

$$\int_0^t \left(\cot(\pi\tau) - \frac{1}{\pi\tau} \right) d\pi\tau = \ln \frac{\sin(\pi\tau)}{\pi\tau} \bigg|_0^t = \ln \left[\frac{\sin(\pi t)}{\pi t} \right]$$

and

$$\int_0^t \sum_{n=1}^{\infty} \frac{-2\tau}{n^2 - \tau^2} d\tau = \sum_{n=1}^{\infty} \ln \left(1 - \frac{t^2}{n^2} \right) = \ln \left[\prod_{n=1}^{\infty} \left(1 - \frac{t^2}{n^2} \right) \right]$$

from which, finally,

$$\frac{\sin(\pi t)}{\pi t} = \prod_{n=1}^{\infty} \left(1 - \frac{t^2}{n^2} \right).$$

as well pull out the standard tools of harmonic analysis and look a little deeper in the spectral properties of the interpolated signal. The quintessential property of the sinc function is its being the bandlimited function par excellence, with a symmetric compact support around zero and a flat magnitude over the support. Assuming the usual time base of $T = 1$, the sinc's support is between $-\pi$ and π as shown by the Fourier transform

$$\mathfrak{F}[\mathrm{sinc}\,(t)] = \mathrm{rect}\left(\frac{\Omega}{2\pi}\right);$$

increasing the time base just shrinks the spectral support, and vice versa.

It stands to reason that a sinc-interpolated signal, which is just a linear combination of (delayed) sincs, will be in turn a bandlimited signal. This is indeed true for any signal of practical interest and actually very easy to show for signals that are absolutely summable (which, as announced before, is the case we're in anyway); for these signals, after plugging (7) into the Fourier analysis formula, one can interchange integration and summation with impunity and prove the result.

Sinc-interpolated signals are therefore bandlimited signals whose spectral support is determined solely by the interpolation time base; the natural question now is whether any bandlimited signal can be harmlessly sampled into a discrete-time sequence. We know the answer to be yes since we have all heard of the sampling theorem, but here it's about the journey and not the destination. Our preferred route is to show that the space of π-bandlimited functions forms a Hilbert space and that sampling is just the result of a basis expansion. The way to proceed is to first build an orthonormal basis; for this purpose consider the family

$$\varphi^{(n)}(t) = \mathrm{sinc}\,(t-n), \quad n \in \mathbb{Z}.$$

By noting that $\varphi^{(n)}(t) = \varphi^{(0)}(t-n)$, orthogonality can be proved first by rewriting the inner product as a convolution

$$\begin{aligned}\langle \varphi^{(n)}(t), \varphi^{(m)}(t)\rangle &= \langle \varphi^{(0)}(t-n), \varphi^{(0)}(t-m)\rangle \\ &= \int_{-\infty}^{\infty} \mathrm{sinc}\,(\tau) \\ &\quad \times \mathrm{sinc}((m-n)-\tau)d\tau \\ &= (\mathrm{sinc} * \mathrm{sinc})(m-n)\end{aligned}$$

and then by applying the convolution theorem for continuous-time functions to compute its value in $(m-n)$

$$\begin{aligned}\langle \varphi^{(n)}(t), \varphi^{(m)}(t)\rangle &= \mathfrak{F}^{-1}\left[\mathrm{rect}^2\left(\frac{\Omega}{2\pi}\right)\right]_{t=m-n} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\Omega(n-m)}d\Omega \\ &= \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases}\end{aligned}$$

so that $\{\varphi^{(n)}(t)\}_{n\in\mathbb{Z}}$ is indeed orthonormal. To have a Hilbert space, we should also prove the completeness of the space of π-bandlimited functions, as explained in the previous section; unfortunately, that relies on the proof of completeness for the continuous-time Fourier basis, which is long and rather technical and which we are therefore forced to skip in this article. Nonetheless, armed with our (putative) Hilbert space and its orthonormal basis, we can now pick an arbitrary π-bandlimited function $x(t)$ and formally compute its basis expansion coefficients

$$\begin{aligned}\langle \varphi^{(n)}(t), x(t)\rangle &= \langle \varphi^{(0)}(t-n), x(t)\rangle && (9)\\ &= (\varphi^{(0)} * x)(n) && (10)\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{rect}\left(\frac{\Omega}{2\pi}\right) \\ &\quad \times X(j\Omega)e^{j\Omega n}d\Omega && (11)\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(j\Omega)e^{j\Omega n}d\Omega && (12)\\ &= x(n); && (13)\end{aligned}$$

in the derivation we have first rewritten the inner product as a convolution and then applied the convolution theorem; the penultimate line is simply the inverse Fourier transform of $X(j\Omega)$ calculated for $t = n$. Remarkably, the nth basis expansion coefficient is just the sampled value of $x(t)$ at $t = n$. To close the loop once and for all, we need only consider the generic orthonormal basis reconstruction formula (8)

$$\begin{aligned}x(t) &= \sum_{n=-\infty}^{\infty} \langle \varphi^{(n)}(t), x(t)\rangle\, \varphi^{(n)}(t) \\ &= \sum_{n=-\infty}^{\infty} x(n)\mathrm{sinc}\,(t-n), \qquad (14)\end{aligned}$$

which yields the interpolation formula (7), and then we're back to the beginning and the merry-go-round is ready for another spin.

AUTHORS

Paolo Prandoni (paolo.prandoni@epfl.ch) is a cofounder and scientific director of Quividi, an audience measurement company in Paris, France.

Martin Vetterli (martin.vetterli@epfl.ch) is a professor of communication systems and the vice-president for International Affairs at EPF Lausanne, Switzerland.

REFERENCES

[1] M. Unser, "Sampling—50 years after Shannon," *Proc. IEEE,* vol. 88, no. 4, pp. 569–587, Apr. 2000.

[2] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[3] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*. New York: McGraw-Hill, 2001.

[4] E. A. Lee and P. Varaiya, *Structure and Interpretation of Signals and Systems*. Reading, MA: Addison Wesley, 2002.

[5] J. H. McClellan, R. W. Shafer, and M. A. Yoder. *DSP First: A Multimedia Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[6] R. Nowak, R. G. Baraniuk, E. Cands, and M. Vetterli, "Compressive sampling," *IEEE Signal Processing Mag.,* vol. 25, no. 2, pp. 12–13, 2008.

[7] E. Maeland, "On the comparison of interpolation methods," *IEEE Trans. Med. Imag,* vol. 7, no. 3, pp. 213–217, Sept. 1988.

[8] T. Blu, P. Thvenaz, and M. Unser, "Complete parameterization of piecewise-polynomial interpolation kernels," *IEEE Trans. Image Process.,* vol. 12, no. 11, pp. 1297–1309, Nov. 2003.

[9] D. Hairer and G. Wanner, *Analysis by Its History*. New York: Springer-Verlag, 1996.

[10] G. Strang, *Linear Algebra and Its Applications*. Pacific Grove, CA: Brooks Cole, 2005.

[11] N. Young, *An Introduction to Hilbert Space*. Cambridge, U.K.: Cambridge Univ. Press, 1988.

[12] M. Vetterli, J. Kovačević, and V. K. Goyal. (2009). *The World of Fourier and Wavelets: Theory, Algorithms and Applications* [Online]. Available: http://www.fourierandwavelets.org

[13] P. Prandoni and M. Vetterli, *Signal Processing for Communications*. Lausanne: EPFL/CRC Press, 2008.

SP