Crash Course in DSP

January 30, 2025

1 Signal Decomposition Using Sines & Cosines

1.1 Linear Regression

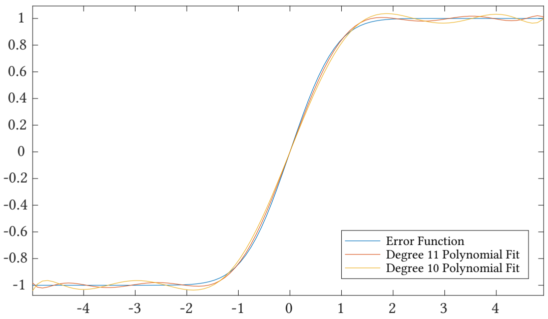

Figure 1: Polynomial interpolation of the error function (integral of Gaussian).

You're probably familiar with the concept of linear regression, in which we try to find a "best fit" line to match some data we have collected. Microsoft Excel calls this best fit line a trendline. We don't need to limit ourselves to a linear best-fit: we can also use higher order polynomials (like quadratics and cubics, etc). In general, the polynomial fit function $g(t)$ can be expressed as:

$$g(t) = a_0 t^0 + a_1 t^1 + a_2 t^2 + ...$$

Where the a_ks are chosen to fit our empirically-gathered data. We can use summation notation to express this same equation in shorthand:

$$g(t) = \sum_{k=0}^{M-1} a_k t^k$$

People have proposed various methods for calculating the a_k coefficients: the Taylor/McLaurin series, least squares, etc. In any case, we are expressing an unknown function $g(t)$ in terms of a sum of known functions (monomials t^k). We call the known functions the "basis functions," and we say that we are expanding $g(t)$ in the basis of monomials. Figure 1 shows polynomial interpolation of the "error function," which is the integral of the Gaussian (e^{-x^2})[1]. In the figure, I have plotted two different polynomial interpolations: one of degree 10, and one of degree 11. There are a couple of shortcomings with polynomial interpolation that make them badly behaved in a lot of important cases, all of which arise from the fact that the individual terms in the polynomial—x, x^2, x^3, etc.—are similar to one another.

Table 1.1 shows the values of the coefficients for each polynomial in Figure 1. While there are some similarities—even-numbered terms have coefficients of zero, for example—these are clearly two different polynomials. Importantly, we cannot just truncate the degree 11 polynomial by one coefficient to get the optimal coefficients for the degree 10 polynomial. The truncated degree 11 polynomial diverges a lot from the degree 10 polynomial. So if we want to change the number of terms in our polynomial interpolation, we need to recalculate the coefficients on each term.

In this lecture series, we are going to expand functions in a different basis set (the basis of sines and cosines, not monomials). This Fourier basis has the property that each term in the expansion is independent, so adding and removing higher-order terms does not affect the coefficients on any of the other terms. This independence or orthogonality property is the crux of digital filtering. We will expand a function in the Fourier basis then truncate the sum to remove noise which is usually present in the higher-order terms. The noise-reduced signal can then be reconstructed from the truncated sum.

1.2 Fourier Series

[1] Since there is no closed-form representation of the error function, we might want to use a polynomial interpolation to quickly evaluate it or manipulate it.

The idea of the Fourier Series is similar to linear regression, except that we use sines and cosines as the basis functions instead of monomials. There are a lot of good reasons for doing this which we can discuss later. First, let's figure out how to calculate a_k coefficients for the Fourier Series. Before we do that though, we need a couple of tricks:

Trick 1: You can take the scalar product of two periodic functions by multiplying them together and integrating over one period:

$$\langle f(t), g(t) \rangle = \int_T f(t)g(t)dt$$

A bit more explanation The scalar product of functions is kind of like the dot product of vectors: it is a measure of similarity between the two functions. If the scalar product is large positive, it means that the functions are very similar. If it is zero or close to zero it means that the functions are not similar. Functions (or vectors) that have a scalar product of zero are said to be *orthogonal*.

Term	Deg 11	Deg 10
0	0	0
1	1.03	0.964
2	0	0
3	-0.221	-0.162
4	0	0
5	0.029	0.015
6	0	0
7	-0.002	-6.3×10^{-4}
8	0	0
9	6.58×10^{-6}	9.7×10^{-6}
10	0	0
11	-8.42×10^{-7}	

Table 1: Coefficients of the polynomials plotted in Figure 1.

Trick 2: Sines (and cosines) that have different frequencies are orthogonal. For example, we can say that:

$$\langle sin(2\pi t), sin(4\pi t) \rangle = \int_{-\pi}^{\pi} sin(2\pi t) sin(4\pi t) = 0$$

See §7.1 for a proof.

Trick 3: Euler's Identity Some smart person noticed a striking similarity in the Taylor Series for $sin\theta$, $cos\theta$, and e^θ:

$$sin\theta = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{\theta^{2n-1}}{(2n-1)!} = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + ...$$

$$cos\theta = \sum_{n=0}^{\infty} (-1)^{n} \frac{\theta^{2n}}{(2n)!} = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + ...$$

$$e^\theta = \sum_{n=0}^{\infty} \frac{\theta^n}{n!} = 1 + \theta + \frac{\theta^2}{2!} + \frac{\theta^3}{3!} + \frac{\theta^4}{4!} + \frac{\theta^5}{5!} + ...$$

Notice that the Taylor series for e^θ contains the same terms as the Taylor series for $sin\theta$ and $cos\theta$, but the singns on the alternating terms are incorrect. We can get the signs to alternate in the Taylor Series expansion of the exponential function by multiplying its argument by the imaginary number i:

$$e^{i\theta} = \sum_{n=0}^{\infty} \frac{(i\theta)^n}{n!} = 1 + i\theta - \frac{\theta^2}{2!} - \frac{i\theta^3}{3!} + \frac{\theta^4}{4!} + \frac{i\theta^5}{5!} - \frac{\theta^6}{6!} - \frac{i\theta^7}{7!} + ...$$

Which yields the Euler identity:

$$e^{i\theta} = cos\theta + isin\theta \tag{1}$$

2 Calculating the Fourier Series Coefficients

2.1 A Simplified Version of the Fourier Series

We are claiming that like linear regression, we can expand any arbitrary function in the basis of sines and cosines:

$$g(t) = \sum_{k=0}^{M-1} a_k sin(2\pi kft) = a_0 sin(0t) + a_1 sin(2\pi ft) + a_2 sin(4\pi ft) + a_3 sin(6\pi ft) + ...$$

Each term in this sum is a sine wave that oscillates at a multiple of the fundamental frequency f. Figure 2 shows the first four terms in this expansion with $f = 1$. To make this expansion work, we need to calculate the expansion coefficients.

$$g(t) = \sum_{k=0}^{M-1} a_k sin(2\pi k f t)$$

Now, we will invoke **Trick 2**, multiplying both sides by a sine with frequency $2\pi f m$ and integrating over one period:

$$\int_T g(t) sin(2\pi m f t) dt = \int_T \sum_{k=0}^{M-1} a_k sin(2\pi k f t) sin(2\pi m f t) dt = \sum_{k=0}^{M-1} a_k \int_T \left(sin(2\pi k f t) sin(2\pi m f t)\right) dt$$

According to **Trick 2**, the integral on the right hand side is zero unless $m = k$, so we can eliminate all but one terms from the sum.

$$\int_T g(t) sin(2\pi m f t) dt = a_m \int_T sin^2(2\pi m t) dt$$

The right-hand-side integral is equal to $T/2$:

$$a_m = \frac{2}{T} \int_T g(t) sin(2\pi m f t) dt \tag{2}$$

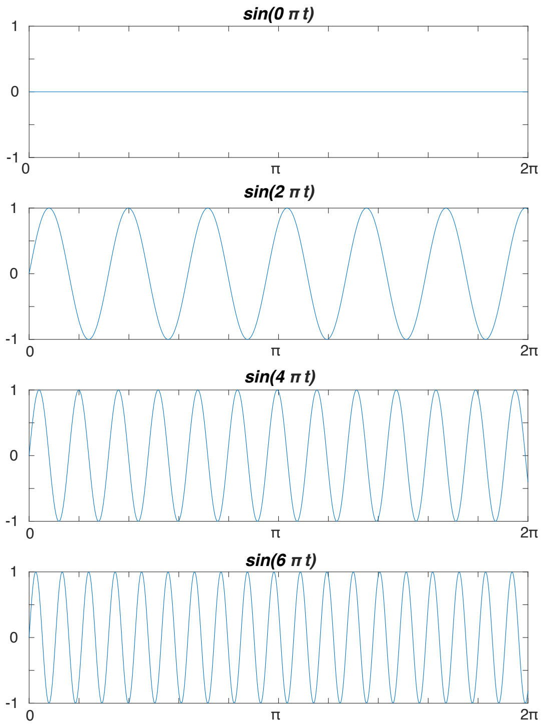

Figure 2: The first four basis functions in the Fourier series with $f = 1$.

Example: Square Wave

$$\begin{aligned} a_m &= \frac{2}{T} \int_T s(t) sin(mt/T) dt \\ &= \frac{1}{\pi} \int_0^{\pi} sin(mt) dt - \frac{1}{\pi} \int_{\pi}^{2\pi} sin(mt) dt \end{aligned}$$

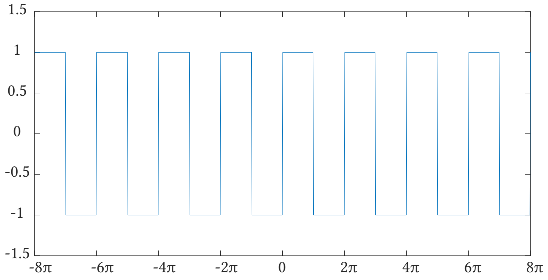

Figure 3: Square wave.

$$\begin{aligned}
a_1 &= \frac{1}{\pi}\int_0^{\pi} sin(t)dt - \frac{1}{\pi}\int_{\pi}^{2\pi} sin(t)dt \\
&= \frac{1}{\pi}\left[(-cos(t))\Big|_0^{\pi} + (cos(t))\Big|_{\pi}^{2\pi}\right] \\
&= \frac{1}{\pi}\left[2+2\right] = \frac{4}{\pi}
\end{aligned}$$

$$\begin{aligned}
a_2 &= \frac{1}{\pi}\int_0^{\pi} sin(2t)dt - \frac{1}{\pi}\int_{\pi}^{2\pi} sin(2t)dt \\
&= \frac{1}{2\pi}\left[(-cos(2t))\Big|_0^{\pi} + (cos(2t))\Big|_{\pi}^{2\pi}\right] \\
&= \frac{1}{2\pi}\left[-(1-1)+(1-1)\right] = 0
\end{aligned}$$

$$\begin{aligned}
a_3 &= \frac{1}{\pi}\int_0^{\pi} sin(3t)dt - \frac{1}{\pi}\int_{\pi}^{2\pi} sin(3t)dt \\
&= \frac{1}{3\pi}\left[(-cos(3t))\Big|_0^{\pi} + (cos(3t))\Big|_{\pi}^{2\pi}\right] \\
&= \frac{1}{3\pi} - \left[cos(3\pi) - cos(0)\right] + \left[cos(6\pi) - cos(3\pi)\right] \\
&= \frac{1}{3\pi}\left[-(-1-1)+1-(-1)\right] \\
&= \frac{4}{3\pi}
\end{aligned}$$

$$\begin{aligned}
a_5 &= \frac{1}{\pi}\int_0^{\pi} sin(5t)dt - \frac{1}{\pi}\int_{\pi}^{2\pi} sin(5t)dt \\
&= \frac{1}{5\pi}\left[(-cos(5t))\Big|_0^{\pi} + (cos(5t))\Big|_{\pi}^{2\pi}\right] \\
&= \frac{1}{5\pi} - \left[cos(5\pi) - cos(0)\right] + \left[cos(10\pi) - cos(5\pi)\right] \\
&= \frac{1}{5\pi}\left[-(-1-1)+1-(-1)\right] \\
&= \frac{4}{5\pi}
\end{aligned}$$

$$\begin{aligned}
a_m &= \frac{1}{\pi}\int_0^{\pi} sin(mt)dt - \frac{1}{\pi}\int_{\pi}^{2\pi} sin(mt)dt \\
&= \frac{1}{m\pi}\left[(-cos(mt))\Big|_0^{\pi} + (cos(mt))\Big|_{\pi}^{2\pi}\right] \\
&= \frac{1}{m\pi} - \left[cos(m\pi) - cos(0)\right] + \left[cos(2m\pi) - cos(m\pi)\right] \\
&= \frac{1}{m\pi}\left[-2cos(m\pi)+1+cos(2m\pi)\right] \\
&= \begin{cases} 4/m\pi & m\ odd \\ 0 & m\ even \end{cases}
\end{aligned}$$

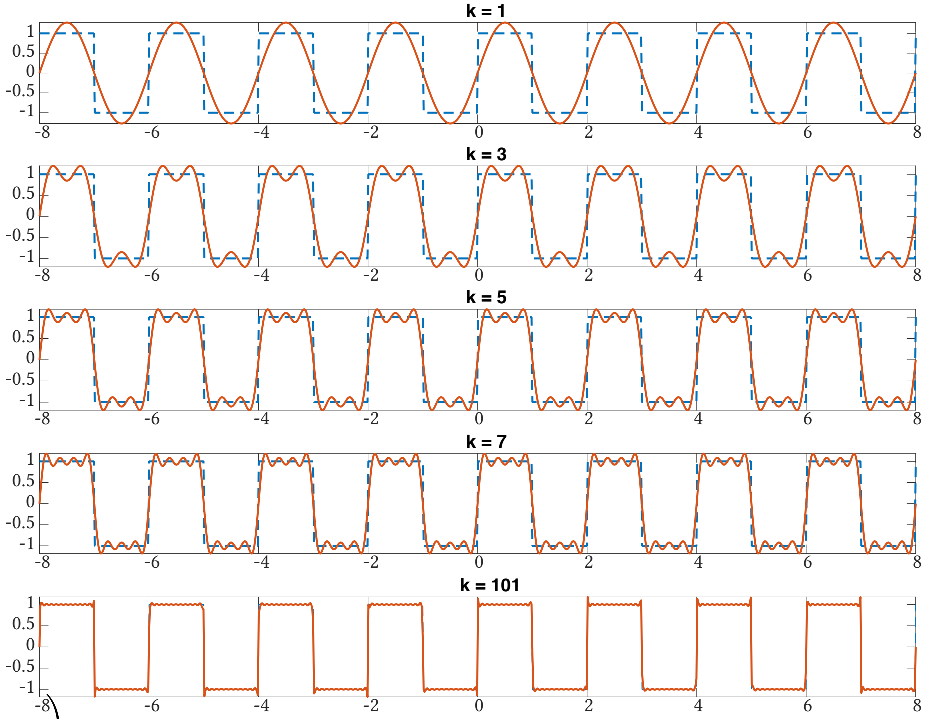

Figure 4: Squre wave.

Example: Sawtooth Wave We will compute the Fourier Series coefficients for the 2π-periodic sawtooth wave $s(t)$ shown in Figure 5 by starting with Equation 4. One period of this function goes from $[-\pi, \pi]$, and its value on that interval is $s(t) = t/\pi$.

$$\begin{aligned} a_m &= \frac{2}{T}\int_T s(t) sin(2\pi mt/T)dt \\ &= \frac{1}{\pi}\int_{-\pi}^{\pi} (t/\pi) sin(mt)dt \end{aligned}$$

Using integration by parts, let $u = t$, $du = dt$, $dv = sin(\frac{2\pi mt}{T}) = sin(mt)$, and $v = -\frac{1}{m}cos(mt)$.

$$\begin{aligned} a_m &= \frac{1}{\pi^2}\left(uv\Big|_{-\pi}^{\pi} - \int_{-\pi}^{\pi} vdu \right) \\ &= \frac{1}{\pi^2}\left(-\frac{t}{m}cos(mt)\Big|_{-\pi}^{\pi} + \frac{1}{m}\int_{-\pi}^{\pi} cos(mt)dt \right) \\ &= \frac{1}{\pi^2}\left(-\frac{\pi}{m}cos(m\pi) - \frac{\pi}{m}cos(-m\pi) \right) \\ &= -\frac{2}{m\pi}cos(m\pi) \end{aligned}$$

Note that $cos(m\pi)$ alternates between $+1$ and -1. If we want to expand the sawtooth function using the a_m coefficients calculated above:

$$s(t) = \frac{2}{\pi}sin(t) - \frac{1}{\pi}sin(2t) + \frac{2}{3\pi}sin(3t) - \frac{2}{4\pi}sin(4t) + \frac{2}{5\pi}sin(5t) - \frac{2}{6\pi}sin(6t) + ...$$

Figure 6 shows plots of approximations of $s(t)$ where this sum has been truncated to a few terms. As we add more terms to the sum, the approximation of the original sawtooth wave becomes better. If we include an infinite number of terms in the sum, the Fourier Series approximation will perfectly match the ideal sawtooth wave.

Another way to vizualize the Fourier Series is by plotting only the values of a_m, the coefficients of the basis functions in our sum. This kind of plot shows the relative contributions of each frequency component to the approximation. Figure 7 shows the values of the Fourier Series coefficients a_m for our sawtooth wave. This is what we call the *frequency spectrum* of the function $s(t)$.

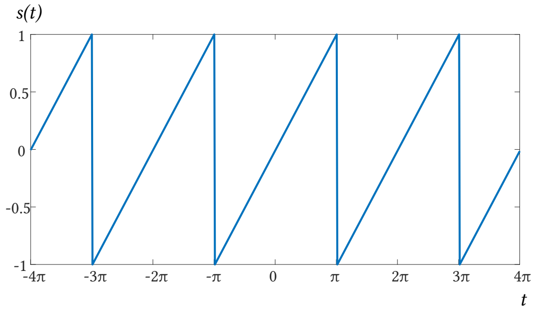

Figure 5: Sawtooth wave.

In the frequency spectrum of $s(t)$, the lower frequency components are dominant, meaning that they are larger in absolute value and they make a greater contribution to the overall approximation. If we want to save memory, we can truncate the sum and save only a subset of the a_m coefficients. We would want to make sure to save the largest-valued coefficients because those are the ones that make the most contribution to the overall sawtooth wave approximation.

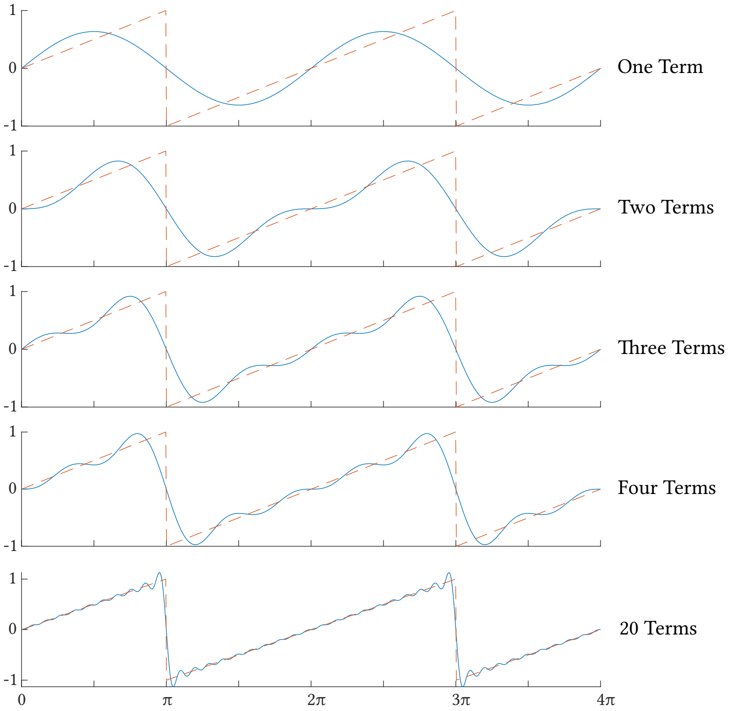

Figure 6: Sawtooth wave.

2.2 Fourier Series for General Functions

In the last section, we represented our target function as a sum of sine waves only. We could do this because our target function (the sawtooth wave) is—like the sine function—antisymmetric about $x = 0$. We call these kinds of antisymmetric functions *odd functions*.

There are also even functions—like the cosine function—that are symmetric about $t = 0$. Even functions can be represented as a sum of cosines only (no sines).

In general, most functions are not symmetric or antisymmetric about $t = 0$. In order to represent a general function in the Fourier Series, we need both sines and cosines.

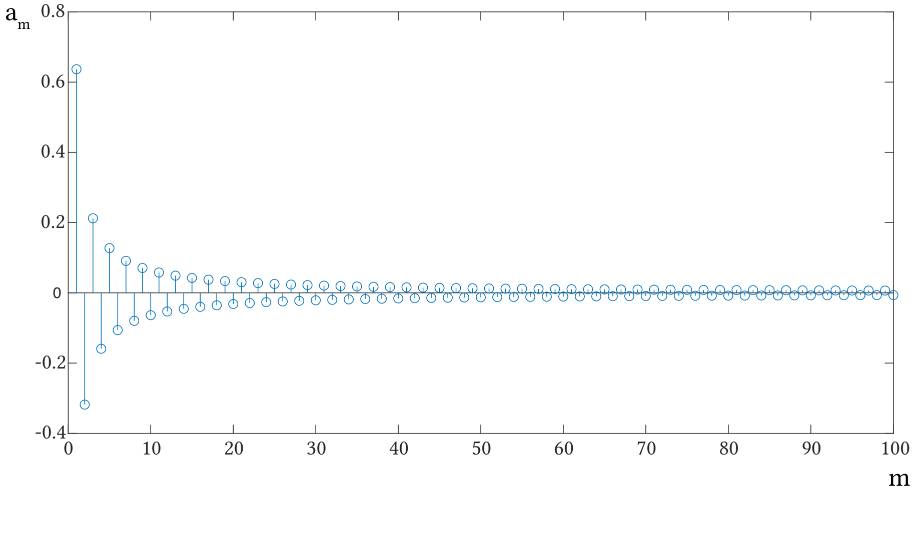

Figure 7: Fourier Series coefficients for the sawtooth wave approximation.

$$g(t) = \sum_{m=0}^{\infty} a_m sin(2\pi mt/T) + b_m cos(2\pi mt/T)$$

We will take advantage of the following two trig identities:

$$cos(t) = \frac{e^{it}+e^{-it}}{2} \qquad sin(t) = \frac{e^{it}-e^{-it}}{2i}$$

```
clear, close all;
N = 20;  % Max number of approximation coefficients
a = 1:N;
n = 1:N;
a = -(2./(pi*n)).*cos(n*pi); % Calculate approximation coeffs as a vector.

% Generate the sawtooth wave. This is not used to calculate anything, just
% to compare our approximation to the ideal sawtooth wave in graphs.
t = linspace(0,4*pi,800);
s(1:200) = linspace(0.,1.,200);
s(201:600) = linspace(-1.,1., 400);
s(601:800) = linspace(-1.,0,200);

p = 1;
f = zeros(1,length(t)); % f holds our approximation of the sawtooth wave
figure;
for k = 1:length(a)
    f = f + a(k) * sin((k)*t); % Add a term to f
    if ((k >= 1) && (k <=4)) || (k == length(a))
        % Plot the first four and last approximations of the sawtooth wave
        subplot(5,1,p);
        hold on;
        plot(t/pi,f);
        plot(t/pi,s,'--');
        p = p + 1;
    end
end

% Plot the frequency spectrum
figure;
stem(a);
```

Figure 8: MATLAB code used to generate plots for the sawtooth wave approximation.

$$
\begin{aligned}
g(t) &= \sum_{m=0}^{\infty} a_m sin(2\pi mt/T) + b_m cos(2\pi mt/T) \\
&= \sum_{m=0}^{\infty} a_m \left(\frac{e^{i2\pi mt/T} - e^{-i2\pi mt/T}}{2i} \right) + b_m \left(\frac{e^{i2\pi mt/T} + e^{-i2\pi mt/T}}{2} \right) \\
&= \sum_{m=0}^{\infty} \frac{a_m}{2i} e^{i2\pi mt/T} - \frac{a_m}{2i} e^{-i2\pi mt/T} + \frac{b_m}{2} e^{i2\pi mt/T} + \frac{b_m}{2} e^{-i2\pi mt/T} \\
&= \sum_{m=0}^{\infty} \frac{-a_m i}{2} e^{i2\pi mt/T} - \frac{-a_m i}{2} e^{-i2\pi mt/T} + \frac{b_m}{2} e^{i2\pi mt/T} + \frac{b_m}{2} e^{-i2\pi mt/T} \\
&= \sum_{m=0}^{\infty} \left(\frac{b_m - a_m i}{2} \right) e^{i2\pi mt/T} + \left(\frac{b_m - a_m i}{2} \right) e^{-i2\pi mt/T}
\end{aligned}
$$

We can now split these two terms in the sum into two separate sums and enforce the rule that $b_n = b_{-n}$ and $a_n = -a_{-n}$.

$$
\begin{aligned}
g(t) &= \sum_{m=1}^{\infty} \left(\frac{b_m - a_m i}{2} \right) e^{i2\pi mt/T} + \sum_{m=-\infty}^{0} \left(\frac{b_{-m} - a_{-m} i}{2} \right) e^{i2\pi mt/T} \\
&= \sum_{m=1}^{\infty} \left(\frac{b_m - a_m i}{2} \right) e^{i2\pi mt/T} + \sum_{m=-\infty}^{0} \left(\frac{b_m - a_m i}{2} \right) e^{i2\pi mt/T} \\
&= \sum_{m=-\infty}^{\infty} \left(\frac{b_m - a_m i}{2} \right) e^{i2\pi mt/T} \\
&= \sum_{m=-\infty}^{\infty} c_m e^{i2\pi mt/T}
\end{aligned}
$$

The final form of the equation gives the standard form in which people usually express the Fourier Series. The coefficients c_m are computed by finding the inner product between

2.3 Inverse Transform

We've already kind of talked about this, but it's worth making it explicit. A couple of facts about the Fourier Transform:

- The frequency spectrum of a signal is a complex-valued function that represents its frequency components. The "forward transform" is the calculation of the complex-valued c_m coefficients from the time-domain signal.
- The frequency spectrum of a signal contains a unique representation of the signal, meaning that we can uniquely reconstruct the time domain signal from its frequency spectrum. This reconstruction is called the *inverse transform.* The inverse transform is the calculation of the time-domain signal from the set of complex-valued c_m coefficients.

Forward transform:

$$c_m = \frac{1}{T} \int_T g(t) e^{-i2\pi nt/T} dt \tag{3}$$

Inverse transform:

$$g(t) = \sum_{m=-\infty}^{\infty} c_m e^{i2\pi mt/T} \tag{4}$$

2.4 Different Kinds of Fourier Transforms

In Equation 4, we are enforcing the rule that the frequency components of our signal must be discrete multiples of the fundamental. This imposes the assumption that our time domain signal is periodic. What if we convert the sum in Equation 4 to an integral? Doing so will allow us to frequency components in the frequency spectrum of any frequency, not just multiples of the fundamental. This gives us a continuous frequency spectrum.

Property Name	**Equation**
Linearity	$Af(t) + Bg(t) \overset{\mathscr{F}}{\longleftrightarrow} AF(\Omega) + BG(\Omega)$
Time Shift	$f(t - t_0) \overset{\mathscr{F}}{\longleftrightarrow} e^{-i\Omega t_0}F(\Omega)$
Modulation (Frequency Shift)	$e^{-i\Omega_0 t}f(t) \overset{\mathscr{F}}{\longleftrightarrow} F(\Omega - \Omega_0)$
Conjugation	$F(\Omega) = F^*(-\Omega)$ iff $f(t)$ is real-valued
Differentiation	$\frac{d}{dt}f(t) \overset{\mathscr{F}}{\longleftrightarrow} i\Omega F(\Omega)$
Convolution	$f(t) * g(t) \overset{\mathscr{F}}{\longleftrightarrow} F(\Omega)F(\Omega)$ $f(t) * g(t) \overset{\mathscr{F}}{\longleftrightarrow} F(\Omega)F(\Omega)$

2.4.1 Fourier Transform

The Fourier Transform is a modification of the Fourier Series in which the frequency domain is considered to be continuous:

$$G(\omega) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi tf/T}dt \tag{5}$$

$$g(t) = \int_{-\infty}^{\infty} G(\omega)e^{i2\pi tf/T}d\omega \tag{6}$$

In general, functions that are periodic in one domain are discrete in the other domain, and functions that are aperiodic in one domain are continuous in the other domain.

Fourier Transform • Continuous and Aperiodic in Time • Continuous and Aperiodic in Frequency	**Fourier Series** • Continuous and Periodic in Time • Discrete and Aperiodic in Frequency
Discrete Time Fourier Transform • Discrete and Aperiodic in Time • Continuous and Periodic in Frequency	**Discrete Fourier Transform** • Discrete and Periodic in Time • Discrete and Periodic in Frequency

Although these transforms deal with different kinds of functions (discrete/continuous and periodic/aperiodic), they generally accomplish the same task of converting between a real-valued time-domain signal and a complex-valued frequency spectrum. In digital signal processing, we work exclusively with the discrete Fourier Transform because it is discrete in both domains. This allows us to represent the values of the signal in an array.

3 Properties of the Fourier Transform

3.1 Computing the Fourier Transform of a Periodic Signal

Generally, if we have a periodic signal in the time domain, we would use the Fourier Series to represent the signal in the frequency domain. But signal analysis using the Fourier Series can be nonintuitive because the frequency domain is parameterized by m—the x-axis of the Fourier Series. The parameter m tells us which basis function we are using. It does not tell us the frequency of the basis function, which is usually what we care about. If we see a feature we are interested in on the Fourier Series frequency spectrum, it can be difficult to determine what frequency that feature corresponds to. The Fourier Transform is more useful because the x-axis represents frequency in Hertz.

$$x(t) \xleftrightarrow{\text{FS};\Omega_0} X[k] = \frac{1}{T}\int_0^T x(t)e^{-i\Omega_0 kt}dt$$

$$X(\Omega) = 2\pi \sum_{k=-\infty}^{\infty} X[k]\delta(\Omega - k\Omega_0) \tag{7}$$

where $\Omega_0 = \frac{2\pi}{T}$.

3.2 Convolution

Convolution of two functions $g(t)$ and $h(t)$ is defined in the following way:

$$g(t) * h(t) = \int_{-\infty}^{\infty} g(\tau)h(t-\tau)dt\tau$$

Let's take the Fourier transform if this convolution and see what happens:

$$\begin{aligned} g(t) * h(t) &= \int_{-\infty}^{\infty} g(\tau)h(t-\tau)dtd\tau \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\tau)h(t-\tau)d\tau e^{i2\pi ft}dt \end{aligned}$$

Now we will substitute $w = t - \tau$, $dw = dt$, $t = \tau + w$:

$$\begin{aligned} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(\tau)h(w)e^{i2\pi f(\tau+w)}dtd\tau \\ &= \left(\int_{-\infty}^{\infty} x(\tau)e^{i2\pi f\tau}d\tau\right)\left(\int_{-\infty}^{\infty} h(w)e^{i2\pi fw}dw\right) \\ &= G(f)H(f) \end{aligned}$$

So convolution of the time-domain functions causes multiplication of the frequency-domain functions. The same is true in reverse: convolution of the frequency domain functions is the same as multiplication of the time-domain functions.

4 Sampling

4.1 Dirac Delta Function

Dirac's delta function is defined as a limiting case of the pulse $p(t)$ shown in Figure 9. The total area of the rectangle $p(t)$ is 1. Dirac's delta function is the function you get in the limit as $\Delta \to 0$: it has a width of 0, infinite height, and a total area of 1.

4.1.1 Sifting Property

The sifting property basically says that Dirac's delta function multiplied by any other arbitrary function $f(t)$ under an integral "picks out" the value of $f(t)$:

$$\begin{aligned} \int_{-\infty}^{\infty} \delta(t)f(t)dt &= \int_{-\infty}^{\infty} \delta(t)f(0)dt \\ &= f(0)\int_{-\infty}^{\infty} \delta(t)dt \\ &= f(0) \end{aligned}$$

Multiplying by a delta function is the mathematical equivalent of sampling a signal at $t = 0$. When we sample a signal an uniform time intervals, we are multiplying my a train of impulses at equally spaced intervals.

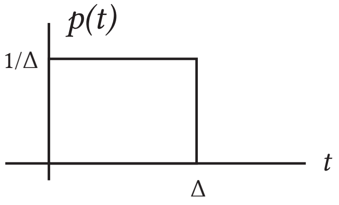

Figure 9: Square pulse function.

Example: Convolution with a Shifted Delta An extension of the sifting property:

$$\begin{aligned} \delta(t-a) * f(t)dt &= \int_{-\infty}^{\infty} \delta(\tau-a)f(t-\tau)d\tau \\ &= f(t-a) \end{aligned}$$

Example: Train of Pulses Consider the train of pulses $s(t)$ shown in Figure 10. Find the frequency spectrum of this function. We will begin by calculating the Fourier Series coefficients for $s(t)$.

$$s(t) = \sum_{l=-\infty}^{\infty} \delta(t - lT)$$

$$a_m = \frac{1}{T}\int_{-T/2}^{T/2} \delta(t) e^{-i\Omega_0 m t} dt = \frac{1}{T}(1)$$

Now plug in the Fourier Series coefficients into Equation 7:

$$S(\Omega) = \frac{2\pi}{T}\sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_0)$$

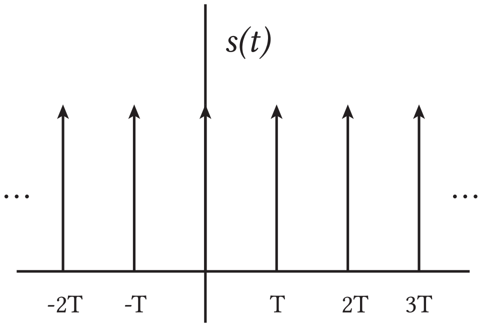

Figure 10: Train of pulses $s(t)$.

The pulse train is one of the very few functions that has the same form in the time and frequency domains. However, the height and spacing of the impulses changes.

When we sample a continuous time signal $f(t)$, we are multiplying it by a train of impulses $s(t)$ to pick out the individual values of the signal at the instants of sampling. The resulting waveform is a *discrete-time signal*, meaning that it only has values at discrete times on the x-axis (see Figure 13 for an example of such a signal). These discrete values can be stored in an array in the computer's memory, and we will discuss techniques of processing them in the next section.

4.2 Consequences of Sampling

Unfortunately, this process of sampling has some undesirable properties that we need to think about. To see what happens, let's take the sampled signal $s(t) \times f(t)$:

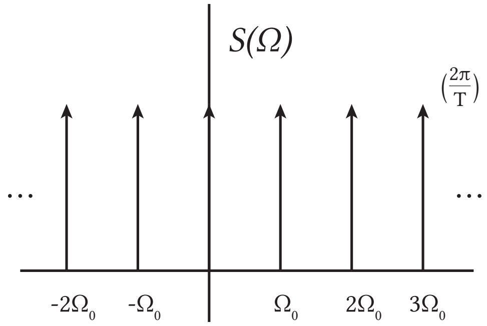

Figure 11: The frequency spectrum of the train of pulses $s(t)$.

$$\mathscr{F}\left\{s(t)f(t)\right\} = S(\Omega) * F(\Omega)$$

What is happening here is we are taking the frequency spectrum of our original signal $F(\Omega)$ and convolving it with the periodic pulse train shown in Figure 11. We showed above that convolution of an arbitrary function $g(t)$ with a shifted impulse $\delta(t-a)$ causes the function $g(t)$ to be shifted right by a:

$$g(t) * \delta(t - a) = g(t - a)$$

But the frequency spectrum of our pulse train $S(\Omega)$ contains infinitely many equally spaced impulses:

$$S(\Omega) = \sum_{a=-\infty}^{\infty} \delta(\Omega - a\Omega_0)$$

$$\begin{aligned}
G(\Omega) * S(\Omega) &= G(\Omega) * \sum_{a=-\infty}^{\infty} \delta(\Omega - a\Omega_0) \\
&= \sum_{a=-\infty}^{\infty} G(\Omega) * \delta(\Omega - a\Omega_0) \\
&= \sum_{a=-\infty}^{\infty} G(\Omega - a\Omega_0)
\end{aligned}$$

Which gives us the frequency spectrum G replicated at intervals of Ω_0, shown in Figure 12.

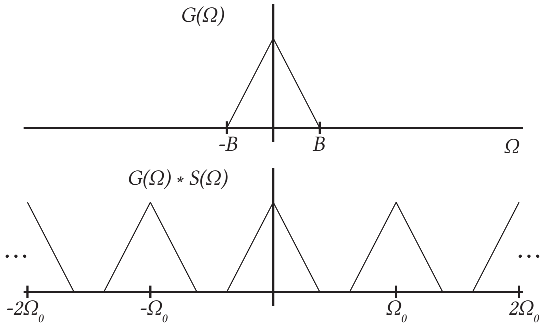

Figure 12: Frequency spectrum of a continuous-time signal $G(\Omega)$ and its discrete-time analog $G(\Omega) * S(\Omega)$.

4.2.1 The Nyquist Theorem / Shannon Sampling Theorem

I have drawn Figure 12 such that the bandwidth of the continuous-time signal is much smaller than the sampling frequency Ω_0. Everyting will be fine as long as $B < \Omega_0/2$. If B becomes larger than $\Omega_0/2$, the frequency replicas will overlap with one another, distorting the sampled signal. This overlap is called *aliasing*. Probably the most recognizable form of aliasing you've likely seen is in car commercials where the car wheel appears to spin backwards. This happens because the camera's frame rate (Ω_0) is lower than the rotation speed of the wheel (B). The wheel goes through more than one revolution per frame, making it appear to spin backwards.

The admonishion to keep the sampling frequency greater than twice the bandwidth of the analog signal is called the *Nyquist Theorem* or the *Shannon Sampling Theorem*. In practice, we often put a simple analog filter called an antialiasing filter between our analog signal and our analog-to-digital converter to enforce the sampling theorem.

5 Digital Filtering

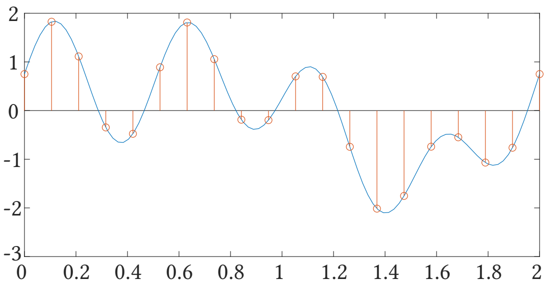

Figure 13: A continuous-time signal and its samples.

In the world of digital filtering, we don't use integrals to compute inner products of functions to find the Fourier Series/Fourier Transform coefficients. Instead, we use numerical methods to calculate the similarity between our digitized signal and the Fourier basis.

First, we sample a signal using an analog-to-digital converter and store the samples in an array. We must be careful to ensure that our sampling obeys the Nyquist theorem. The samples must be taken at uniform time intervals for all this stuff to work.

We then build a basis set, which is a set of complex sinusoids of the form $e^{i2\pi ft}$, sampled at the same time intervals as our ADC-acquired signal. To compute each Fourier coefficient, we calculate the vector inner product between the sampled signal and each of the discrete basis functions. Usually we use the same number of basis functions as samples, and usually that number is a power of 2. This whole thing works out to a matrix-vector multiplication:

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ \dots \\ S_{N-1} \end{bmatrix} = \begin{bmatrix} e^{i2\pi 0/N} & e^{i2\pi 0/N} & e^{i2\pi 0/N} & \dots & e^{i2\pi 0/N} \\ e^{i2\pi 0/N} & e^{i2\pi 1/N} & e^{i2\pi 2/N} & \dots & e^{i2\pi (N-1)/N} \\ e^{i2\pi 0/N} & e^{i2\pi 2/N} & e^{i2\pi 4/N} & \dots & e^{i2\pi 2(N-1)/N} \\ e^{i2\pi 0/N} & e^{i2\pi 3/N} & e^{i2\pi 6/N} & \dots & e^{i2\pi 3(N-1)/N} \\ & & \dots & & \\ e^{i2\pi 0/N} & e^{i2\pi (N-1)/N} & e^{i2\pi 2(N-1)/N} & \dots & e^{i2\pi (N-1)^2/N} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ \dots \\ s_{N-1} \end{bmatrix}$$

The column vector on the right-hand-side of this equation is the time-domain input signal $s[n]$. The left-hand-side of this equation is the discrete-time frequency spectrum of $s[n]$.

5.1 Noise

6 Compressive Sensing

We can represent the FFT operation as a matrix-vector multiplication:

$$F = \Psi f \tag{8}$$

where Ψ is a matrix in which each row contains samples of the complex sinusoid $e^{-i2\pi ft/T}$on the interval $[0, 2\pi]$. Ψ is a unitary matrix, which means that its conjugate transpose[2] is its inverse:

$$\Psi\Psi^* = I$$

The Nyquist Theorem provides a lower bound on the spacing of these complex exponentials in frequency given the maximum frequency of the continuous-time signal. The higher the frequency of the continuous time analog signal, the smaller spacing of the complex exponentials in Ψ.

However, the Nyquist Theorem assumes that the continuous-time signal is *broadband*: that is, it contains non-negligible components of almost all frequencies. This assumption is often not true. We frequently deal with *sparse* signals in which many components of Ψ are zero or close to zero.

When we eliminate rows of the Ψ matrix, the system in Equation 8 becomes overdetermined, meaning there are more equations than unknowns (or more data points than degrees of freedom). Overdetermined systems are often solved with the method of least squares, which minimizes the square of the deviations between each point and the approximation:

$$\hat{x} = \min_x \sum_k ||x_k - y_k||_1$$

Least squares tends to overpenalize large deviations from the mean and favor solutions where all of the components of the solution have small but nonzero values. This is an undesirable characteristic when we expect solutions to be sparse. To get a sparse solution, we want a way to force many (or most) components to zero while allowing a few components to be larger nonzero values.

6.1 Overdetermined Systems

Step 1

$$f = \Psi c \tag{9}$$

Compute the expansion of $f_{n\times 1}$ in basis $\Psi_{n\times n}$ with loading vector $c_{n\times 1}$, which tells us how much each component of Ψ contributes to the sampled signal. This is the IFFT.

Step 2

$$b = \phi f \tag{10}$$

Where $\phi_{n\times m}$ is a rectangular measurement matrix, which selects a subset of the samples in f. It is basically the identity matrix with some columns removed. Vector b is measurements of the original signal sampled at sparse intervals. Combining Equations 9 and 10:

$$b = \phi\Psi c \tag{11}$$

This is an overdetermined system. We want to solve this system for c subject to the constraint that we minimize $||c||_1$ because that constraint tends to extract sparse solutions.

[2] To get the conjugate transpose, we interchange rows and columns and take the complex conjugate of each element. To get the complex conjugate of a complex exponential, invert the sign of the argument of the exponential: $e^{-ia} \rightarrow e^{+ia}$

```
clear all, close all, clc

n = 5000;
t = linspace(0,1/8,n);
% Signal f is the 'A' tone on touchtone phone
f = sin(1394*pi*t) + sin(3266*pi*t);

figure(1);
subplot(2,1,1);
plot(t,f);

% Plot discrete cosine transform of f.
% Kinda like a real-only FFT
ft = dct(f);
subplot(2,1,2);
plot(t,ft);

% Randomly select m (< n) random samples of f
m = 500;
temp = randperm(n); % randomly permute integers in range 1:n
ind = temp(1:m); % Choose the first m randomly permuted points
tr = t(ind); % tr holds the time values at the m randomly chosen indices
% Evaluate function f at the m randomly chosen indices
fr = sin(1394*pi*tr) + sin(3266*pi*tr);

subplot(2,1,1);
hold on;
scatter(tr,fr);

% Build the basis matrix Psi
Psi = dct(eye(n,n));

% Build A = \phi \Psi, selecting only a subset of the basis vectors in Psi
A = Psi(ind,:);

%% Bad Solution: Use Least Squares to find c.
% x is an estimate of the DCT coefficients of the original signal.
x = pinv(A)*fr';

% Inverse DCT to get time-domain estimate
ls_estimate = dct(x);

figure(2);
subplot(2,1,1);
plot(t,f);
hold on;
plot(t,ls_estimate);
title 'Least Squares Estimate of Signal'

subplot(2,1,2);
plot(t,ft);
hold on;
plot(t,x);
title 'DCT of Least Squares Solution'

```

```
%% Good Solution: Use L1 norm minimization to find c

[x1, FitInfo] = lassoglm(A,fr');
l1_estimate = dct(x1(:,25));

figure(3);
subplot(2,1,1);
plot(t,f);
hold on;
plot(t,l1_estimate);
title 'L1 Estimate of Signal'

subplot(2,1,2);
plot(t,ft);
hold on;
plot(t,x1(:,25));
title 'DCT of L1 Solution'
```

7 Appendix

7.1 Proof that Sines of Different Frequencies are Orthogonal

The period of the lower frequency sinusoid is:

$$\omega_1 = 2\pi f_1 = 2\pi/T_1$$

$$T_1 = 2\pi/\omega_1$$

The period of the higher frequency sine wave is a multiple of $2\pi/\omega_1$. To compute the inner product, we will integrate over T_1 because both sines are periodic on T_1.

$$\begin{aligned}
\langle sin(\omega_1 t), sin(k\omega_1 t)\rangle &= \int_0^{2\pi/\omega_1} sin(\omega_1 t) sin(k\omega_1 t) dt && (12)\\
&= \int_0^{2\pi/\omega_1} \frac{1}{2}\left(cos(\omega_1 t - k\omega_1 t) - cos(\omega_1 t + k\omega_1 t)\right) dt && (13)\\
&= \frac{1}{2}\int_0^{2\pi/\omega_1} cos((1-k)\omega_1 t) dt - \frac{1}{2}\int_0^{2\pi/\omega_1} cos((1+k)\omega_1 t) dt && (14)\\
&= \frac{1}{2}\left[\frac{sin((1-k)\omega_1 t)}{(1-k)\omega_1}\right]_0^{2\pi/\omega_1} - \frac{1}{2}\left[\frac{sin((1+k)\omega_1 t)}{(1+k)\omega_1}\right]_0^{2\pi/\omega_1} && (15)\\
&= \frac{1}{2}\left[\frac{sin((1-k)\omega_1 \frac{2\pi}{\omega_1})}{(1-k)\omega_1} - 0\right] - \frac{1}{2}\left[\frac{sin((1+k)\omega_1 \frac{2\pi}{\omega_1})}{(1+k)\omega_1} - 0\right] \qquad = 0 && (16)
\end{aligned}$$

Since $sin(2\pi k) = 0$.