Fourier Analysis

Looking back at the drum and the vibrating string, we note that we could have represented their motion in two different ways:

1) Traveling waves
For instance, on the string, we can represent its displacement $y(x,t)$ at any time as the sum of a right-moving wave and a left-moving wave

$$y(x,t) = y_R(x-ct) + y_L(x+ct)$$

And finding the displacement of the string at any time is equivalent to a superposition of these two traveling waves. We might call this a time-analysis of the string.

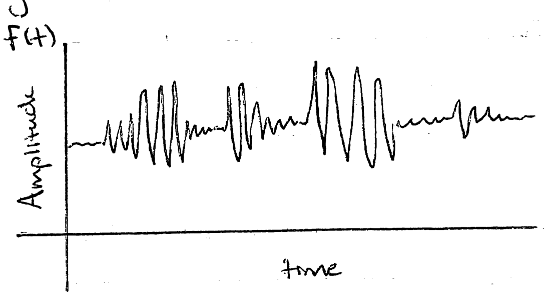
2) Standing waves
We can also represent the displacement of the string in terms of standing waves of different frequencies, and find the displacement as a "recipe" of these standing wave frequencies. "How much" (the amplitude) of each standing wave component we add determines the overall displacement, as

$$y(x,t) = \sum_{n=1}^{\infty} \left(A_n \cos(\omega_n t) + B_n \sin(\omega_n t)\right) \sin(k_n x)$$

We might call this a frequency-analysis of the string.

In theory, either representation should work to find the displacement of the string, but sometimes, one is simpler than the other. The question is, then, is there a systematic way to convert between these two representations?

Another case where this idea sees use is in music recording. Say we have a performer sing into a microphone. What the microphone will pick up is the amplitude or intensity of the singer's voice as a function of time.



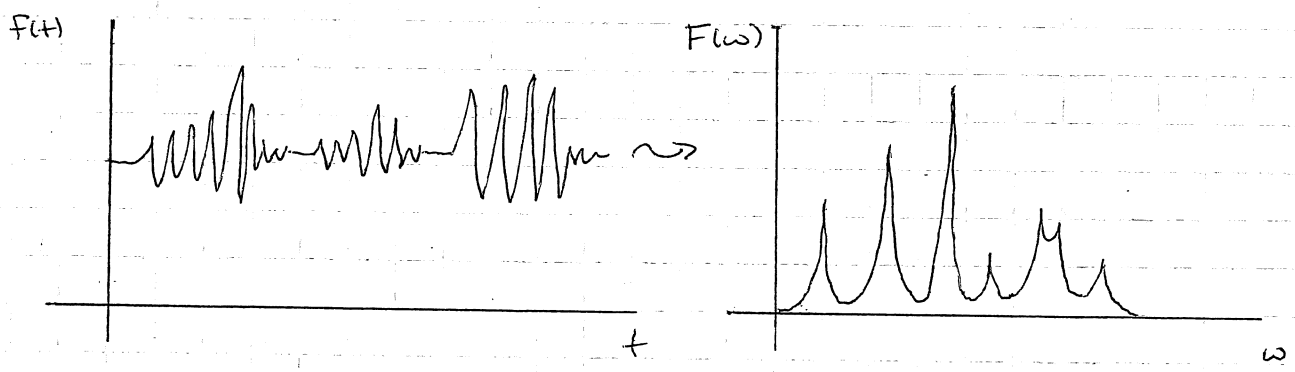
But say that, after they have finished putting their all into that recording, you go back to listen to it and you notice that there's a distinct, high-pitched humming in the background the whole time, and at one point, the singer hits a flat note.

Now you could just have the singer re-record, but alas, they've already left!

How do you fix this? All you've got to work with in the recording is the amplitude. Varying this would make the whole performance louder or quieter, so it wouldn't fix the hum or the flat note.

Instead, you realize you need to look at the recording in terms of the amplitudes of each frequency present. If you can figure out what that high frequency is, you can drop its amplitude to 0, getting rid of the hum. Further you could adjust the frequency of the flat note until it's pitch perfect! But the question remains:

Given only a time-varying signal $f(t)$, can you get a corresponding frequency representation $F(\omega)$? The answer is yes, but it takes a bit of work.



Fourier first found that (almost) any periodic signal of period T could be written as a series of sines and cosines, of the form

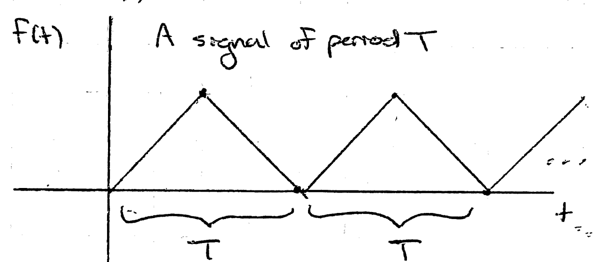
$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2\pi n t}{T}\right) + b_n \sin\left(\frac{2\pi n t}{T}\right) \right)$$

where

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} f(t)\, dt$$

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos\left(\frac{2\pi n t}{T}\right) dt$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin\left(\frac{2\pi n t}{T}\right) dt$$



Or, in terms of complex exponentials (using that $e^{i\theta} = \cos\theta + i\sin\theta$),

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{i2\pi n t}{T}}, \quad c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-\frac{i2\pi n t}{T}} dt$$

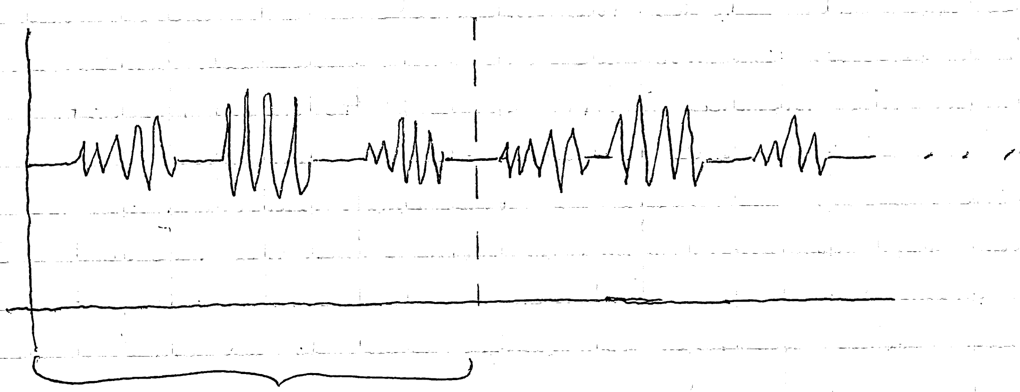
But, remembering that T is the period of the signal, $\frac{2\pi}{T} = \omega_1$ is the fundamental angular frequency of the signal, i.e. the angular frequency at which the entire periodic waveform repeats. Then we can write $f(t)$ as

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_1 t}, \quad c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-in\omega_1 t} dt$$

This says that (almost) any periodic function can be represented as an infinite combination of phasors, rotating at different frequencies $n\omega_1$.

The amount of each phasor is determined by what $f(t)$ is, and so, in a loose sense, we have expressed $f(t)$ in terms of component frequencies $n\omega_1$.

Now, the problem is that, unless it's a pop song, a typical song isn't repetitive enough to be periodic. However, there is a workaround here. Say the length of the song is 3 minutes. Then we might consider as our input function f(t) the amplitude of the song as a function of time, but the song restarts after 3 minutes:



3 minutes

What's easier, and more general though, is to think of a nonperiodic signal as one where the period is infinite i.e., let $T \to \infty$ in our series.
If we do this, our fundamental frequency ω_1 becomes very small, since $\omega_1 = \frac{2\pi}{T}$.
Further, the difference in frequencies for each consecutive phasor gets smaller, since

$$\Delta\omega = (n+1)\omega_1 - n\omega_1 = \omega_1.$$

Let's call this difference $\Delta\omega$.
Then we have for our series that

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_1 t}$$

$$= \sum_{n=-\infty}^{\infty} \left\{ \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-in\omega_1 t} dt \right\} e^{in\omega_1 t}$$

$$= \sum_{n=-\infty}^{\infty} \left\{ \frac{1}{2\pi} \int_{-T/2}^{T/2} f(t) e^{-in\omega_1 t} dt \right\} e^{in\omega_1 t} \Delta\omega$$

As T gets larger, $\omega_1 = \Delta\omega$ gets smaller, so $n\omega_1$ gets smaller as well. In the limit, $n\omega_1$ approaches a continuous variable ω, and our sum becomes an integral of the form

$$f(t) = \int_{-\infty}^{\infty} \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \right\} e^{i\omega t} d\omega$$
$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \right\} e^{i\omega t} d\omega$$

Now the term in braces will be a function of only ω, since it is a definite integral over t. Therefore, let's call this expression $F(\omega)$. Then we have

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} dt,$$

where

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt.$$ (analogous to c_n for periodic signals)

Since $F(\omega)$ is dependent on the form of f(t), we can consider it a representation of the function f in terms of its component frequencies. But this is exactly what we wanted! A way to convert from an expression of a signal as a function of time to one as a function of frequency. We call $F(\omega)$ the Fourier transform of $f(t)$, since it transforms the function from a representation in one domain (time) to another domain (frequency). $f(t)$ is then the inverse Fourier transform of $F(\omega)$. Assuming $f(t)$ expresses the amplitude of the signal over time, $F(\omega)$ expresses the amplitude of the frequencies present in the signal.

In the case of our singer, we now know what we can do. Take the time-based signal we received from the recording, Fourier transform it, reduce the amplitude of the high-pitched humming frequency, then inverse Fourier transform this edited signal to get back a recording without the humming!
We can fix the flat note in a similar way. The only problem is that this note is not necessarily flat everywhere in the song, so we don't want to Fourier transform the whole signal and edit the pitch, or we will change it everywhere it appears in the song.
Rather, we should only transform the function within a small "window" of time, adjust the frequency of the flat note in that interval, then inverse transform.
You may already see how the Fourier transform could have many uses in music, including the two we've focused on thus far. The first is an example of what's called equalization, where we adjust the amplitudes of certain frequencies or frequency ranges to get a desired sound (e.g. bass boost, cut out mids, etc.)
The second is a precursor to the Autotune technology, which pitch shifts all notes to the nearest semitone to produce a perfectly-pitched track, or, when used in more extreme ways, distort the human voice in a distinct way.
Some other uses of the FT include

1) Electronic Tuners
2) Synthesizers
3) Shazam
4) File compression
5) Digital Effects Pedals

We should note here that the FT used in these applications is not exactly the one we gave, but rather one that takes in <u>samples</u> of a signal as input, not the full signal itself. This is to save on both space and computation time. This then leads to the question of what is the minimum number of samples needed to fully reproduce the original signal? In most applications, this is answered by the Nyquist-Shannon sampling theorem.