ENGIN 100: Music Signal Processing

LAB #3: Discrete Spectral Analysis

Professor Andrew E. Yagle

Dept. of EECS, The University of Michigan, Ann Arbor, MI 48109-2122

I. ABSTRACT

This lab will teach you how to use a tool central to signal processing: spectral analysis of signals using the discrete Fourier transform (FFT). The Background section presents: (a) the physics of vibrating strings; (b) the Fourier series expansion of a periodic signal; (c) the sampling theorem; and (d) computation of Fourier series coefficients. The goals of this lab are: (1) To gain the ability to use Matlab's FFT command to compute Fourier series coefficients from samples of a periodic signal; (2) To use these computed coefficients to filter a noisy signal; (3) To compute these coefficients for simple sinusoids and plucked string signals to determine their spectra. These have obvious applications in the music synthesizer and transcriber projects.

II. BACKGROUND

There is no way to present this material without a lot of mathematics. I have put the algebra into 3 separate appendices, at the end of this lab. This means you can skip them on first reading of the lab, and then plow through them later, with a mug of non-decaf coffee or tea. You won't be tested on them (whew!).

Appendix A is a simple 1-D analysis of the physics of a vibrating string. You should be able to follow it if you have the basics of the first month of Math 115 (differential calculus) down. The point of this material is that there are physical reasons for a vibrating string in particular, and music in general, to sound like a fundamental sinusoid plus harmonics. Bernoulli first proposed this model for music, and we will follow him.

Appendix B derives the sampling theorem for bandlimited periodic signals: A signal can be reconstructed perfectly from its samples if it is sampled at greater than twice its maximum frequency. This result is why digital signal processing, CDs, DVDs, and pretty much everything digital exists: without it, digital could never be as good as analog (ugh!). In fact, the periodicity is unnecessary, but the proof for non-periodic signals requires much more math (specifically, the Fourier transform and continuous-time impulses). But if we let the period be one century or more, we won't be around when it starts to repeat!

Appendix C derives explicit formulae for computing the Fourier coefficients from sampled data. The math here isn't nearly as bad as it looks (really!), but it is tough wading. But it sure beats solving a huge set of simultaneous linear equations! These are the formulae we will use for spectral analysis of music.

III. Fourier Series Expansions of Periodic Signals ("All You Need is Trig")

From Appendix A, we can model the sound of a plucked string as (here $T = 2L/a$)

$$x(t) = C_0 + \sum_{k=1}^{M} C_k \cos(2\pi kt/T - \theta_k) = A_0 + \sum_{k=1}^{M} A_k \cos(2\pi kt/T) + B_k \sin(2\pi kt/T), \tag{1}$$

for some constants $\{A_k, B_k, C_k, \theta_k\}$ related to each other by (remember from Lab #1)

$$A_k = C_k \cos\theta_k; \quad B_k = C_k \sin\theta_k; \quad C_k = \sqrt{A_k^2 + B_k^2}; \quad \tan\theta_k = B_k/A_k. \tag{2}$$

What does $x(t)$ look like? Clearly it is periodic (it repeats in time) with period=T:

$$x(t) = x(t+T) = x(t+2T)\ldots = x(t-T) = x(t-2T)\ldots$$

In fact *any* periodic function that occurs in the real world can be represented as a sum of sinusoids with frequencies that are integer multiples of the "frequency"=1/period of the function. We need to include the integer zero here since the signal may have a DC (constant) component. But music signals don't have DC components, so in the sequel we assume there is no DC component. Also note that B_0=0 since sin(0)=0.

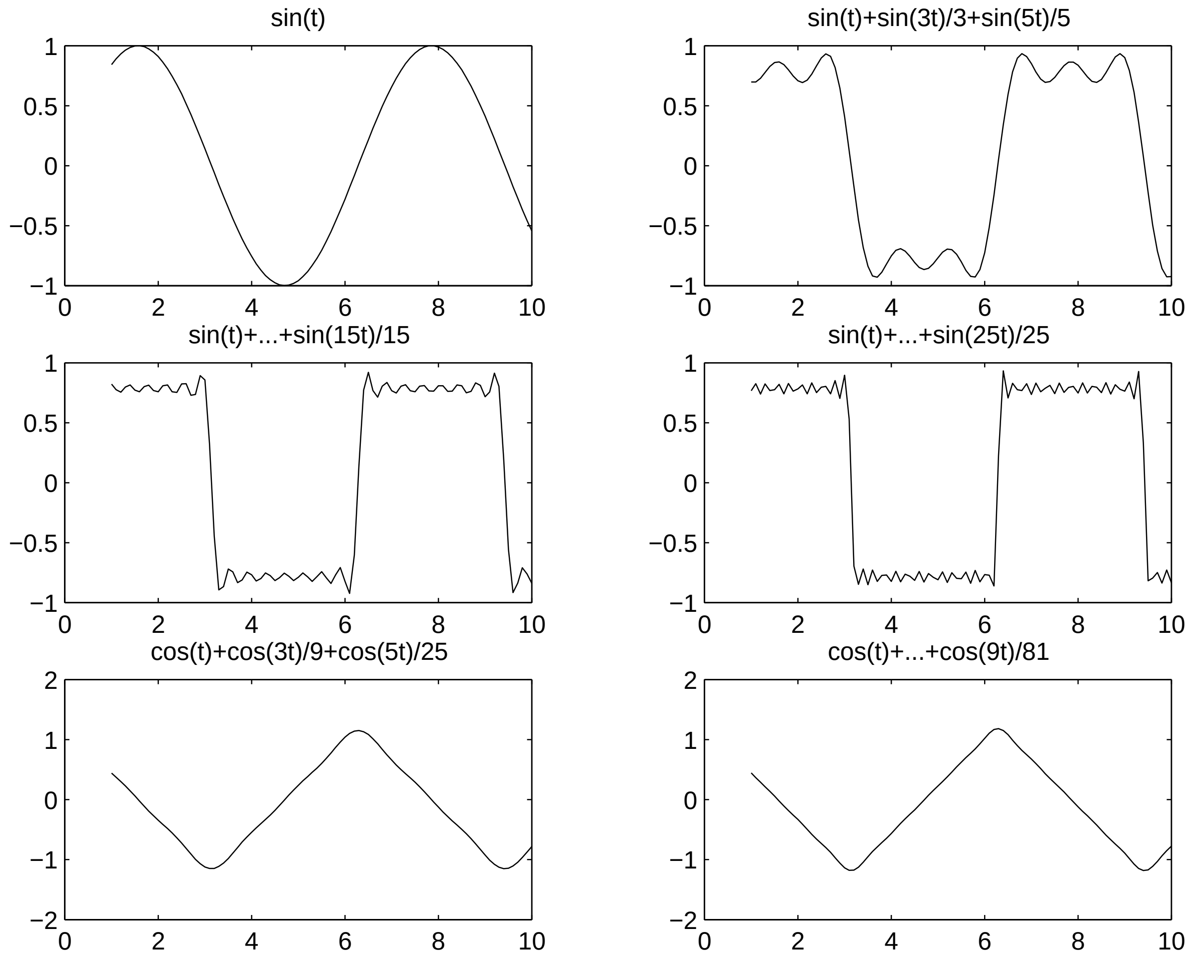
If we let $M \to \infty$ we can handle non-real-world signals like square waves. However, this gets into convergence issues that are very messy. In 1807, at a meeting of the Paris Acadeny, Jean Fourier claimed any periodic function could be expanded as a (possibly infinite) sum of sinusoids. Lagrange stood up and said he was wrong, and an argument ensued that lasted for years (they didn't have TV then). It turns out that there are exceptions, but they are (as mathematicians put it) "pathological," so don't worry about them.

What does this mean for Engin 100? Musical signals are periodic and can be expressed as a weighted sum of sinusoidal signals whose frequencies are integer multiples of the "frequency"=1/period of the musical signal. If there is only one sinusoid, the musical signal is a pure tone. If there is more than one sinusoid, the extra sinusoids are called *harmonics* of the *fundamental* sinusoid, which would be the pure tone (T=period):

$$x(t) = \underbrace{C_1 \cos(2\pi t/T - \theta_1)}_{\text{FUNDAMENTAL: } \frac{1}{T}\text{Hz}} + \underbrace{C_2 \cos(4\pi t/T - \theta_2)}_{\text{HARMONIC: } \frac{2}{T}\text{Hz}} + \underbrace{C_3 \cos(6\pi t/T - \theta_3)}_{\text{HARMONIC: } \frac{3}{T}\text{Hz}} + \ldots \tag{3}$$

- The constants C_k produce the *timbre* (pronounced "tam-ber") of the sound;
- The harmonics are also called *partials* (note they change if the period does);
- $\{A_k, B_k, C_k, \theta_k\}$ can be computed easily from *samples* of the signal $x(t)$.

The next page shows two simple examples of Fourier series converging to square and triangle waves. Note that the triangle wave converges faster than the square wave, since it has no discontinuities.


sin(t)
sin(t)+sin(3t)/3+sin(5t)/5
sin(t)+...+sin(15t)/15
sin(t)+...+sin(25t)/25
cos(t)+cos(3t)/9+cos(5t)/25
cos(t)+...+cos(9t)/81
1
0.5
0
−0.5
−1
2
−2
0
2
4
6
8
10

IV. Using Matlab to Compute Fourier Series Coefficients

Matlab's `fft` computes $(A_k–iB_k)$N/2, i=$\sqrt{-1}$. Given N=2L samples `X` of a complete period of $x[n]$,

$$\texttt{L} = \texttt{length(X)}/2; A_k = \texttt{real(fft(X))}/\texttt{L}; B_k = -\texttt{imag(fft(X))}/\texttt{L}; C_k = \texttt{abs(fft(X))}/\texttt{L}; \theta_k = \texttt{angle(fft(X))}$$

`fft` is short for Fast Fourier Transform, which is a very efficient algorithm for computing A_k and B_k, especially when N is a power of two. Note that the FFT is an algorithm, not a result–it is uncouth to speak of "FFTing" a data set, although many engineers (incorrectly) do so. Note the following:

- `fft(X)` has the same length as `X`, although there are half as many A_k (M+1) and B_k (M) as there are of $x[n]$ (N=2M+1). The 2^{nd} half of `fft(X)` is $(A_{N-k}+iB_{N-k})$N/2, which is the mirror image of the 1^{st} half.
- So when plotting spectra computed using `fft(X)`, **ONLY PLOT THE FIRST HALF!**.
- Then `X=real(ifft(FX))` computes `X` from `fft(X)`, i.e., it sums the discrete-time Fourier series.
- The 1^{st} value of `fft(X)` is the DC (average) value of `X`. For musical signals this is zero. But for `FX=fft(X)`, `N=length(X)`, use: A_0=`FX(1)/N`; A_k=`2*real(FX(k+1))/N`; B_k=`-2*imag(FX(k+1))/N` for $k = 1 \ldots \frac{N}{2}$.
- Yes, this indexing is confusing. You will get all of this straight by doing the lab exercises below.

A. Simple Example: Interpretation of Matlab's `fft`

A data acquisition system provides data sampled at $1024\frac{\text{SAMPLE}}{\text{SECOND}}$. The data is read into Matlab (using command `fread`) and into a row vector `X`. How do we use Matlab to determine the spectrum of the data?

We run the following Matlab commands, and get the following results:

- >> `length(X)` yields 3072. The duration is (3072 samples)$\left(\frac{1}{1024}\frac{\text{SECOND}}{\text{SAMPLE}}\right)$=T=3 seconds.
- >> `-(2/3072)*imag(fft(X,3072))` yields 3072 numbers $< 10^{-14}$. That is roundoff, meaning it's zero!
- >> `(2/3072)*real(fft(X,3072))` yields zeros **except** at the following indices (locations):
- $\{97, 193, 289, 2785, 2881, 2977\}$. At those indices the values are $\{4, 3, 2, 2, 3, 4\}$, respectively.
- Given all of the above information, compute a sum-of-sinusoids formula for the data.

We interpret these results as follows:

- The duration=T=3 seconds means we take the periodic extension of the 3-second-long data.
- This models the data as being periodic with period=3 seconds, with harmonics at $\{\frac{1}{3}, \frac{2}{3} \ldots\}$ Hertz.
- The final three components are the mirror images of the first three, both in value and location:
- 2785=3072+2–289; 2881=3072+2–193; 2977=3072+2–97. Why do we need to add 2 each time?
- A component at Matlab index K corresponds to frequency (K–1)$\frac{S}{N}$=(K–1)$\frac{1}{T}$ where N=ST.
- We have components at (97–1)/3=32 Hertz. (193–1)/3=64 Hertz. (289–1)/3=96 Hertz. That's all!
- These are all pure cosines, since the B_k are zero (to roundoff; see above).
- The data can therefore be written as $x(t) = 4\cos(2\pi 32t) + 3\cos(2\pi 64t) + 2\cos(2\pi 96t)$.
- Hence the data are actually periodic with period=1/32 second, not 3 seconds.

B. Harder Example: Interpretation of Matlab's `fft`

Matlab includes a sampled train whistle signal. Let's compute its spectrum.

We run the following Matlab commands, and get the following results:

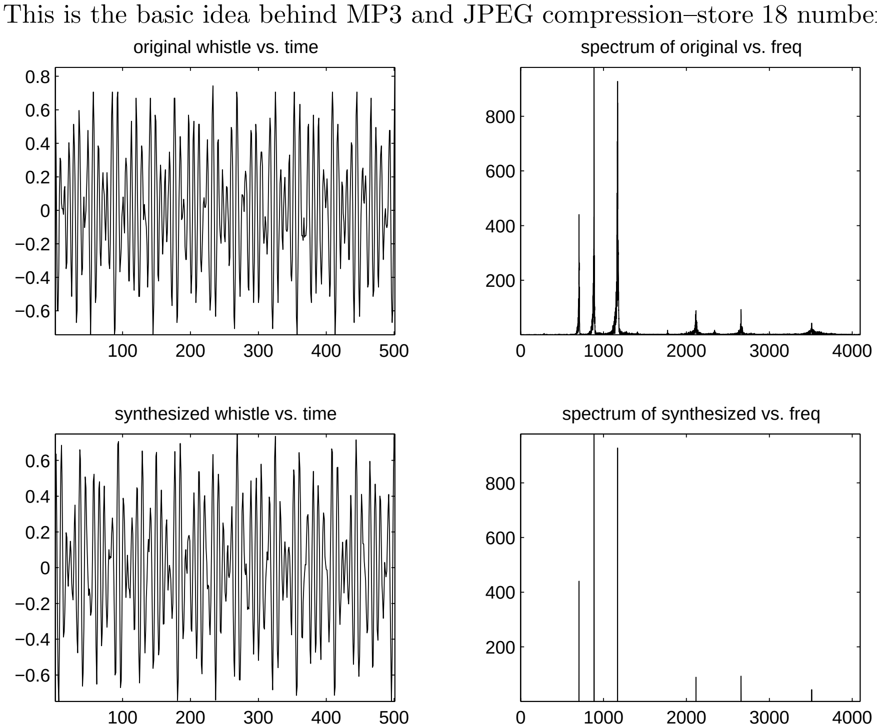
- `>>load train.mat;plot(y);F=fft(y);plot(abs(fft(F(1:round(length(y)/2)))))`
- Remember to plot only the first half of `abs(F)` to get the spectrum! This is a **very** common mistake!
- The 2 upper plots are the signal and its spectrum; the 2 lower plots are the approximation given below.
- `>> length(y)` is 12880. The duration of the signal is (12880 samples)$\left(\frac{1}{8192}\frac{\text{SECOND}}{\text{SAMPLE}}\right)$=T=1.57 seconds.
- The sinusoid depicted by a spike at Matlab index `K` has frequency (K-1)/T=$\frac{K-1}{1.57}$ Hertz.
- The amplitudes and phases can be computed using `>> [2/12880*abs(F) angle(F)]`.
- There are only 6 significantly nonzero components:

Index	K	1109	1394	1840	3326	4180	5521
Hertz	$\frac{K-1}{1.57}$	705	886	1170	2115	2658	3511

You now know the frequencies, and you can read off amplitudes from the plot. The signal is $\frac{2}{12880}\times$:

$$440\cos(2\pi 705t-1.70)+979\cos(2\pi 886t+2.93)+928\cos(2\pi 1170t+1.35)$$
$$+88\cos(2\pi 2115t+1.41)+93\cos(2\pi 2658t+0.84)+43\cos(2\pi 3511t+3.03) \quad (4)$$

- The 1^{st} 3 frequencies are in 5:3 and 5:4 ratios. This is suggestive of chords in the whistle sound.
- The $4^{th}-6^{th}$ frequencies are triple the $1^{st}-3^{rd}$ frequencies. Are these harmonics? I think so.
- Comparing the upper and lower plots shows that just 6 sinusoids represent the signal well.
- This is the basic idea behind MP3 and JPEG compression–store 18 numbers instead of 12880!



V. LAB #3: WHAT YOU HAVE TO DO

- A. Spectra of Simple Signals
  - Plot the following three plots on one page using subplot.
  - Type >> `F=fft(cos(2*pi*137*[0:499]/500));plot(abs(F))`. Examine values of `F` to confirm that this is a sinusoid with frequency 137 Hertz. Note the second peak is a mirror image of the first!
  - Type >> `F=fft(cos(1000*[0:499]/500));plot(abs(F))`. Note the broad base around the peak. Examine values of `F` to confirm that this is a sinusoid with frequency $\frac{1000}{2\pi} \approx 159$ Hertz. Why don't you get just a spike, like you did previously? HINT: Does the sample F have an integral number of periods?
  - Download *plucked.mat* from the course web site. Type >> `load plucked.mat;sound(y);plot(abs(fft(y)))`, a *synthetic* signal. What is its period and fundamental frequency? How many harmonics does it have?
- B. Filtering Noisy Data to (Almost) Eliminate the Noise
  - Plot the following three plots on one page using `subplot`.
  - Download *noisy.mat*. Type >> `load noisy.mat;plot(y)`. It should look like noise.
  - Hidden in this noise is a signal, sampled at $1000\frac{\text{SAMPLE}}{\text{SECOND}}$. All you will be told is:
    * It is periodic with period=0.2 seconds, and it is bandlimited to about 20 Hz.
  - From that information, clean up the signal by filtering out most of the noise.
  - HINTS: How many harmonics does it have, and at what frequencies?
  - Set other frequencies to zero and use `X=real(ifft(F))` where `F=fft(X)`.
  - Plot the original noisy signal, its spectrum, and the cleaned-up signal.
- C. **Spectrogram**-Computing Separate Spectra of Different Intervals of a Single Signal
- Use `subplot` to put the 3 plots from (A) and (B) on one page, and the 2 plots from (C) on another page.
  - 1. Chirp signal
    * Type >> `X=cos([1:1000].^2/1000);plot(X)` Describe this signal. Try listening to it.
    * Type >> `imagesc(abs(fft(reshape(X,40,25)))),colormap(gray)`. See what this is doing?
  - 2. Tonal music
    * Type >> `[Y,FS]=wavread('victorstone.wav');` after downloading the file *victorstone.wav*.
    * Type >> `imagesc(abs(fft(reshape(Y,300,260)))),colormap(gray)`. See what this is doing?
    * Can you read off the relative pitches and durations of the tones from the spectrogram?
    * Could a spectrogram of a musical signal function as a type of musical notation?
  - 3. Removing interference from a signal
    * A Michigan State fan broke into the Engin 100 CTools web site and corrupted "The Victors" by adding to it the MSU fight song! Can you undo this heresy? How can we remove this interference?
    * Type >> `[Z,FS]=wavread('victorsmsu.wav');` after downloading the file *victorsmsu.wav*.
    * Type >> `F=fft(Z);plot(abs(F))` Can you distinguish the UM and MSU fight song spectra?
    * Type >> `imagesc(abs(fft(reshape(Z,300,260)))),colormap(gray)`. How about now?
    * Set some values of `F` to zero, then `sound(real(ifft(F)))`. Did you eliminate MSU?

VI. APPENDIX A: PHYSICS OF VIBRATING STRINGS

The basic instrument of rock and folk music is the guitar. A guitar consists of several strings; plucking these strings to make them vibrate creates the music. So let us examine the basic physics of a horizontal vibrating string. This requires basic differential calculus that you should have seen by now in Math 115.

Let x be horizontal position along a string of length L, $t \geq 0$ be time, $y(x,t)$ be vertical displacement of the string at position x and time t, and T be the tension in the string, which is almost horizontal. T does not vary with x since the change of length of the string is negligible when it is plucked or vibrating.

Recall that the slope of a function $y(x)$ at position x is $\frac{dy}{dx}$. The vertical component of force at position x is $T\sin(\theta(x)) \approx T\tan(\theta(x)) = T\frac{dy}{dx}$ and the vertical component of force at position $x+\Delta$ is $T\frac{dy}{dx} + \frac{d}{dx}[T\frac{dy}{dx}]\Delta$ in the opposite direction. The net force is the difference $T\frac{d^2y}{dx^2}\Delta$, since T is assumed not to vary with x.

This force acts on a mass $\rho\Delta$, where ρ is the linear density (mass per unit length) of the string. Newton's law of motion $F = ma$ in the vertical direction then gives

$$ma = (\rho\Delta)\frac{d^2y}{dt^2} = F = T\frac{d^2y}{dx^2}\Delta \rightarrow \frac{d^2y}{dt^2} = a^2\frac{d^2y}{dx^2}. \quad \textbf{UNITS}: T : \text{mass}\frac{\text{length}}{\text{time}^2}; \rho : \frac{\text{mass}}{\text{length}}; a^2 = \frac{T}{\rho} : \frac{\text{length}^2}{\text{time}^2} \tag{5}$$

where $a^2 = \frac{T}{\rho}$ depends only on the string material itself. The derivatives here should really be *partial* derivatives $\frac{\partial^2 y}{\partial t^2}$ and $\frac{\partial^2 y}{\partial x^2}$; this just means that the other variable is treated as a constant in each case.

Since the ends of the string are fixed at both ends $x = 0$ and $x = L$, the equation and its solution are

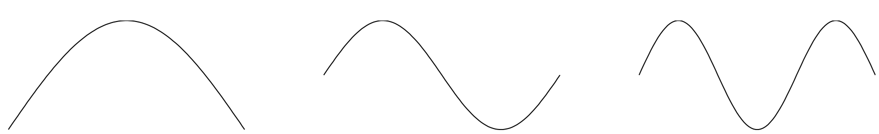
$$\frac{d^2y}{dt^2} = a^2\frac{d^2y}{dx^2}; \quad y(x=0,t) = y(x=L,t) = 0 \rightarrow y(x,t) = \sum_{k=1}^{M} E_k \sin(k\pi x/L)\cos(ak\pi t/L - \theta_k) \tag{6}$$

In fact *all* solutions have the form, although you need a course in partial differential equations to show this.

What does (6) *look* like? If we took a snapshot of the vibrating string at fixed time $t = t_o$ we would see

$$y(x,t_o) = \sum_{k=1}^{M}[E_k\cos(ak\pi t_o/L - \theta_k)]\sin(k\pi x/L), \tag{7}$$

This is a weighted sum of functions that look like (note that the endpoints are always fixed)



These are called *modes of vibration*. Guitar players create a mode by fingering strings.

What does (6) *sound* like? Sound is generated by the time variation

$$y(x_o,t) = \sum_{k=1}^{M}[E_k\sin(k\pi x_o/L)]\cos(k\pi at/L - \theta_k). \tag{8}$$

VII. APPENDIX B: SAMPLING THEOREM

$x(t)$ is real-valued with period=T seconds and maximum frequency $\frac{M}{T}$ Hertz (note that the maximum frequency *must* have this form for some integer M). Then $x(t)$ can be expanded as (1). This means that $x(t)$ is completely specified by 2M+1 constants $\{A_k, B_k\}$ or equivalently $\{C_k, \theta_k\}$. If we can compute those 2M+1 constants, we know $x(t)$ for all t. Now sample $x(t)$ at N=2M+1$\frac{\text{SAMPLE}}{\text{SECOND}}$, so we obtain the samples

$$x[n] = x(t = n\Delta) = x(t = nT/N), \quad n = 1, 2 \dots N. \tag{9}$$

Setting $t = \frac{nT}{N}, n = 1, 2 \dots N$=2M+1 yields N linear equations in N unknowns $\{A_0, A_k, B_k, k = 1 \dots M\}$:

$$x[n] = x(nT/N) = A_0 + \sum_{k=1}^{M} A_k \cos[(2\pi k/T)(nT/N)] + B_k \sin[(2\pi k/T)(nT/N)], \quad n = 1 \dots N \tag{10}$$

We can solve these equations to obtain $\{A_0, A_k, B_k, k = 1 \dots M\}$ from samples $\{x(nT/N), k = 1 \dots N\}$. So we can reconstruct $x(t)$ *exactly* from its samples $x(nT/N)$. Note that we need to sample $x(t)$ at a rate $\frac{N}{T}\frac{\text{SAMPLE}}{\text{SECOND}}$, more than *double* the maximum frequency $\frac{M}{T}$ Hertz. The doubling is because we need *pairs* of constants $\{A_0, A_k, B_k, k = 1 \dots M\}$. This was discovered by Claude Shannon, a UM alumnus. That's his bust outside the EECS building (on the left side as you enter) on the North Campus diag.

A. *Small Illustrative Numerical Example*

Let $x(t)$ be periodic with period $T = 0.001$ and have maximum frequency 1000 Hertz.

$x(t)$ is sampled at 4000$\frac{\text{SAMPLE}}{\text{SECOND}}$. The goal is to reconstruct a formula for the signal.

We observe: x(1/4000)=4; x(2/4000)=1; x(3/4000)=2; x(4/4000)=5. Since the signal is periodic,

$$x(t) = a_0 + a_1 \cos(2\pi t/0.001) + a_2 \cos(4\pi t/0.001) + \dots + b_1 \sin(2\pi t/0.001) + b_2 \sin(4\pi t/0.001) + \dots \tag{11}$$

Since the signal has maximum frequency 1000 Hertz, $x(t) = a_0 + a_1 \cos(2\pi t/0.001) + b_1 \sin(2\pi t/0.001)$.

So the signal is completely determined by three constants: $\{a_0, a_1, b_1\}$. $t = n/4000$ for $n = 1, 2, 3, 4$ gives

$$4 = x(1/4000) = a_0 + a_1 \cos(2\pi \frac{1}{0.001}\frac{1}{4000}) + b_1 \sin(2\pi \frac{1}{0.001}\frac{1}{4000}) = a_0 + a_1 \cos(\frac{1\pi}{2}) + b_1 \sin(\frac{1\pi}{2}) = a_0 + b_1 = 4$$
$$1 = x(2/4000) = a_0 + a_1 \cos(2\pi \frac{1}{0.001}\frac{2}{4000}) + b_1 \sin(2\pi \frac{1}{0.001}\frac{2}{4000}) = a_0 + a_1 \cos(\frac{2\pi}{2}) + b_1 \sin(\frac{2\pi}{2}) = a_0 - a_1 = 1$$
$$2 = x(3/4000) = a_0 + a_1 \cos(2\pi \frac{1}{0.001}\frac{3}{4000}) + b_1 \sin(2\pi \frac{1}{0.001}\frac{3}{4000}) = a_0 + a_1 \cos(\frac{3\pi}{2}) + b_1 \sin(\frac{3\pi}{2}) = a_0 - b_1 = 2$$
$$5 = x(4/4000) = a_0 + a_1 \cos(2\pi \frac{1}{0.001}\frac{4}{4000}) + b_1 \sin(2\pi \frac{1}{0.001}\frac{4}{4000}) = a_0 + a_1 \cos(\frac{4\pi}{2}) + b_1 \sin(\frac{4\pi}{2}) = a_0 + a_1 = 5$$

Solving these four linear equations in three unknowns gives $a_0 = 3; \quad a_1 = 2; \quad b_1 = 1$. So

$$x(t) = 3+2\cos(2\pi 1000t)+1\sin(2\pi 1000t) = 3+\sqrt{5}\cos(2\pi 1000t-\tan^{-1}(1/2)) = 3+2.236\cos(2\pi 1000t-26.6^o).$$

VIII. APPENDIX C: DIRECT COMPUTATION OF FOURIER SERIES COEFFICIENTS

Solving N linear equations in N unknowns is very time-consuming if N is large. Fortunately, we can avoid this by using formulae we now derive. WARNING: Grab a mug of non-decaf coffee or tea before reading!

First, recall the sine and cosine addition formulae

$$\begin{aligned} \sin(x \pm y) &= \sin(x)\cos(y) \pm \cos(x)\sin(y) \\ \cos(x \pm y) &= \cos(x)\cos(y) \mp \sin(x)\sin(y). \end{aligned} \tag{12}$$

Adding and subtracting these gives

$$\begin{aligned} 2\cos(x)\cos(y) &= \cos(x-y) + \cos(x+y) \\ 2\sin(x)\sin(y) &= \cos(x-y) - \cos(x+y) \\ 2\sin(x)\cos(y) &= \sin(x-y) + \sin(x+y). \end{aligned} \tag{13}$$

Second, note that since $\sin(\frac{2\pi k}{N}n)$ is periodic in n with period N and $\sin(-x) = -\sin(x)$ we have

$$\sum_{n=1}^{N} \sin(\frac{2\pi k}{N}n) = \sum_{n=1}^{N/2} \sin(\frac{2\pi k}{N}n) + \sum_{n=-1}^{-N/2} \sin(\frac{2\pi k}{N}n) = \sum_{n=1}^{N/2} [\sin(\frac{2\pi k}{N}n) - \sin(\frac{2\pi k}{N}n)] = 0, \tag{14}$$

where $N/2$ means $\frac{N-1}{2}$ if N is odd (note that $\sin(\frac{2\pi k}{N}\frac{N}{2})$=0 if N is even). Then

$$\begin{aligned} 2\sin(\frac{2\pi k}{N}) \sum_{n=1}^{N} \cos(\frac{2\pi k}{N}n) &= \sum_{n=1}^{N} 2\sin(\frac{2\pi k}{N})\cos(\frac{2\pi k}{N}n) = \sum_{n=1}^{N} [\sin(\frac{2\pi k}{N}(n+1)) - \sin(\frac{2\pi k}{N}(n-1))] \\ = \sum_{n=1}^{N} \sin(\frac{2\pi k}{N}(n+1)) - \sum_{n=1}^{N} \sin(\frac{2\pi k}{N}(n-1)) &= 0 - 0 = 0 \rightarrow \sum_{n=1}^{N} \cos(\frac{2\pi k}{N}n) = 0, 1 \leq k \leq N-1 \end{aligned} \tag{15}$$

after dividing by $2\sin(\frac{2\pi k}{N}) \neq 0$ if $1 \leq k \leq N-1$ and $k \neq N/2$. If $k = N/2$ then (since N is even)

$$\sum_{n=1}^{N} \cos(\frac{2\pi k}{N}n) = \sum_{n=1}^{N} \cos(\pi n) = \sum_{n=1}^{N} (-1)^n = 0. \tag{16}$$

Third, we now have

$$2\sum_{n=1}^{N} \sin(\frac{2\pi i}{N}n)\cos(\frac{2\pi j}{N}n) = \sum_{n=1}^{N} [\sin(\frac{2\pi(i-j)}{N}n) + \sin(\frac{2\pi(i+j)}{N}n)] = 0 - 0 = 0 \text{ even if } i = j. \tag{17}$$

We also now have

$$2\sum_{n=1}^{N} \sin(\frac{2\pi i}{N}n)\sin(\frac{2\pi j}{N}n) = \sum_{n=1}^{N} [\cos(\frac{2\pi(i-j)}{N}n) - \cos(\frac{2\pi(i+j)}{N}n)] = 0 - 0 = 0 \text{ unless } i = j, \tag{18}$$

$$2\sum_{n=1}^{N}\cos(\frac{2\pi i}{N}n)\cos(\frac{2\pi j}{N}n)=\sum_{n=1}^{N}[\cos(\frac{2\pi(i-j)}{N}n)+\cos(\frac{2\pi(i+j)}{N}n)]=0-0=0 \text{ unless } i=j. \tag{19}$$

These three equations are called *orthogonality* conditions, since they state that the inner ("dot") products of $\{\sin(\frac{2\pi i}{N}n)\}$ for two different i are zero, and similarly for $\{\cos(\frac{2\pi i}{N}n)\}$.

Finally (hooray!), setting $t = nT/N$ in (1) gives

$$x[n]=x(t=\frac{nT}{N})=\sum_{k=0}^{M}A_k\cos(\frac{2\pi k}{N}n)+\sum_{k=1}^{M}B_k\sin(\frac{2\pi k}{N}n). \tag{20}$$

This is the *Discrete-Time Fourier Series* (DTFS). Multiplying (20) by $\sin(\frac{2\pi m}{N}n)$ and summing over n gives

$$\begin{aligned}\sum_{n=1}^{N}x[n]\sin(\frac{2\pi m}{N}n)&=\sum_{n=1}^{N}[\sum_{k=0}^{M}A_k\cos(\frac{2\pi k}{N}n)]\sin(\frac{2\pi m}{N}n)+\sum_{n=1}^{N}[\sum_{k=1}^{M}B_k\sin(\frac{2\pi k}{N}n)]\sin(\frac{2\pi m}{N}n)\\&=\sum_{k=0}^{M}A_k\sum_{n=1}^{N}\cos(\frac{2\pi k}{N}n)\sin(\frac{2\pi m}{N}n)+\sum_{k=1}^{M}B_k\sum_{n=1}^{N}\sin(\frac{2\pi k}{N}n)\sin(\frac{2\pi m}{N}n)=NB_m/2\end{aligned} \tag{21}$$

using the orthogonality conditions. Similarly, multiplying (20) by $\cos(\frac{2\pi m}{N}n)$ and summing over n gives

$$\begin{aligned}\sum_{n=1}^{N}x[n]\cos(\frac{2\pi m}{N}n)&=\sum_{n=1}^{N}[\sum_{k=0}^{M}A_k\cos(\frac{2\pi k}{N}n)]\cos(\frac{2\pi m}{N}n)+\sum_{n=1}^{N}[\sum_{k=1}^{M}B_k\sin(\frac{2\pi k}{N}n)]\cos(\frac{2\pi m}{N}n)\\&=\sum_{k=0}^{M}A_k\sum_{n=1}^{N}\cos(\frac{2\pi k}{N}n)\cos(\frac{2\pi m}{N}n)+\sum_{k=1}^{M}B_k\sum_{n=1}^{N}\sin(\frac{2\pi k}{N}n)\cos(\frac{2\pi m}{N}n)=NA_m/2\end{aligned} \tag{22}$$

using the orthogonality conditions derived above. So (**bottom line**) we have the formulae

$\boxed{A_m=\frac{2}{N}\sum_{n=1}^{N}x[n]\cos(\frac{2\pi m}{N}n)}$; $\boxed{B_m=\frac{2}{N}\sum_{n=1}^{N}x[n]\sin(\frac{2\pi m}{N}n)}$, $\quad m=1,2\ldots N$

which we can plug into (1) to get an explicit formula for $x(t)$ for all t.

We need not solve N equations in N unknowns–we can write down the solution directly, thanks to those orthogonality conditions! These formulae compute Fourier series coefficients *directly* from *sampled* data.

A. Small Illustrative Numerical Example, Continued

Continuing the example in Appendix B, we can compute $\{a_0, a_1, b_1\}$ directly:

$$\begin{aligned}a_0&=\frac{1}{4}[4\cos(2\pi\frac{0}{0.001}\frac{1}{4000})+1\cos(2\pi\frac{0}{0.001}\frac{2}{4000})+2\cos(2\pi\frac{0}{0.001}\frac{3}{4000})+5\cos(2\pi\frac{0}{0.001}\frac{4}{4000})=3\\a_1&=\frac{1}{4}[4\cos(2\pi\frac{1}{0.001}\frac{1}{4000})+1\cos(2\pi\frac{1}{0.001}\frac{2}{4000})+2\cos(2\pi\frac{1}{0.001}\frac{3}{4000})+5\cos(2\pi\frac{1}{0.001}\frac{4}{4000})=2\\b_1&=\frac{1}{4}[4\sin(2\pi\frac{1}{0.001}\frac{1}{4000})+1\sin(2\pi\frac{1}{0.001}\frac{2}{4000})+2\sin(2\pi\frac{1}{0.001}\frac{3}{4000})+5\sin(2\pi\frac{1}{0.001}\frac{4}{4000})=1\end{aligned} \tag{23}$$

in agreement with Appendix B. This replaces solving the linear system of 4 equations in 3 unknowns.