Signal Processing and Linear Systems1

Lecture 8: Sampling and The Discrete Fourier Transform

Nicholas Dwork
www.stanford.edu/~ndwork

So far we have used the Fourier Transform to analyze systems.

Our results have been analytical (we've had to the do the work by hand).

We are now going to figure out how to have a computer do much of the work for us.

We will see that the computer will be able to approximate the Fourier Transform, and that we can characterize the errors due to the approximation well.

Outline

Sensors and Sampling

The Discrete Fourier Transform

Characterizing Errors

Sensors gather data from physical systems

Piezoelectric resistors change resistance based on pressure.

Thermistors change resistance based on temperature.

Antennas convert the presence of electromagnetic waves into voltage.

Photodiodes convert visible light into electrical current.

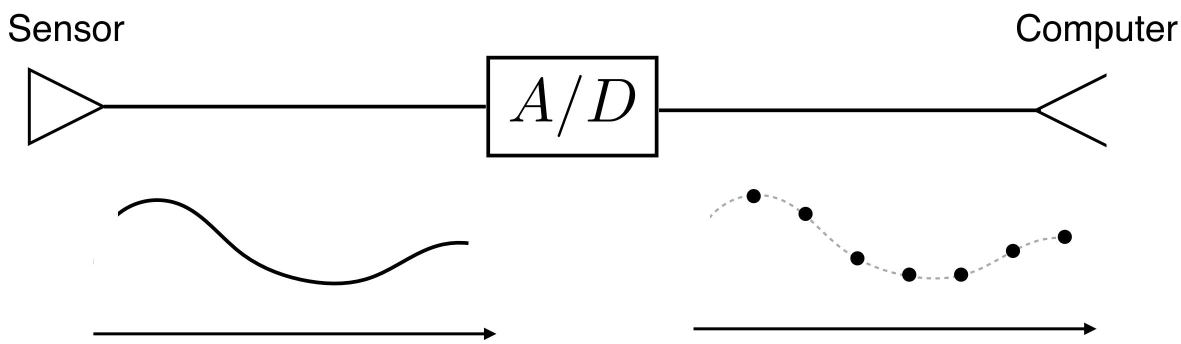
A microphone converts sound waves into voltage.

Many many more.

How do we get the data from a sensor into a computer?

Analog to Digital Converters

The sensors we've discussed output a continuous signal.



The A/D converts a continuous function into a series of numbers that are then stored in computer memory.

Sampling

$$\boldsymbol{f}_n = f(n\Delta)$$

Δ is the spacing between samples.

The set of values of f are called samples: $\{f(n\Delta) : n \in \mathbb{Z}\}$

An ideal A/D converter evaluates the input function at specific times.

Suppose we are given samples of a function. Can we reconstruct the function perfectly?

You would guess that we couldn't; we're throwing away so much information, aren't we?

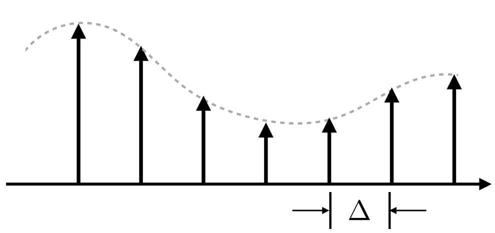
But it turns out that (in a somewhat special case), we can!

Consider the set of samples $\boldsymbol{f} = (\ldots, f(-2\Delta), f(-\Delta), f(0), f(\Delta), f(2\Delta), \ldots)$.

We know the values of the function and the times that the samples were collected.

We are going to do something weird. (This will turn out to be an incredibly ingenious idea!)

Construct an impulse train with the weights of the function.



$$f_s(x) = \sum_{n=-\infty}^{\infty} f(x)\,\delta(x - n\Delta)$$

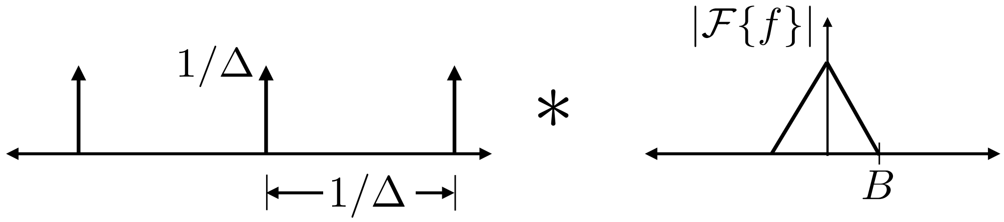
$$f_s(x) = \sum_{n=-\infty}^{\infty} f(x)\,\delta(x - n\Delta)$$

What is the spectrum of f_s? $f_s = \text{Ш}_\Delta\, f$ where $\text{Ш}_\Delta(x) = \sum_{n=-\infty}^{\infty} \delta(x - n\Delta)$.

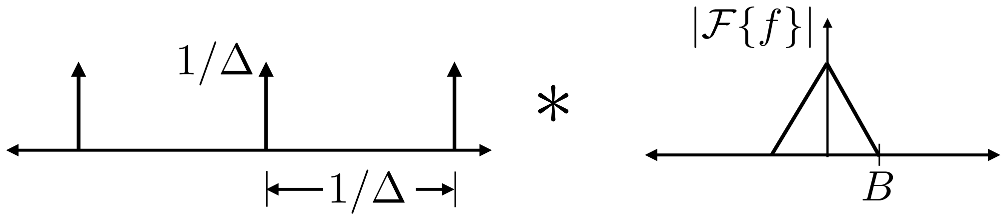
$$\mathcal{F}\{f_s\} = \mathcal{F}\{\text{Ш}_\Delta\, f\} = \mathcal{F}\{\text{Ш}_\Delta\} * \mathcal{F}\{f\} = \frac{1}{\Delta}\text{Ш}_{1/\Delta} * \mathcal{F}\{f\}.$$

Suppose the bandwidth of f is *B*.

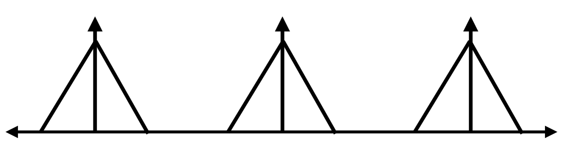
$\mathcal{F}\{f_s\} =$



$\mathcal{F}\{f_s\} =$

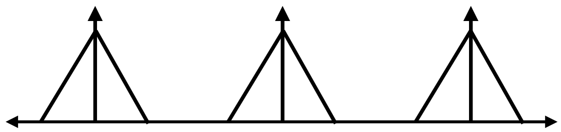


=



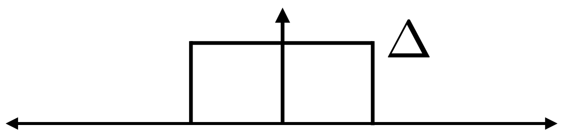
Do the spectra overlap?

If $1/\Delta > 2B$, then they don't.

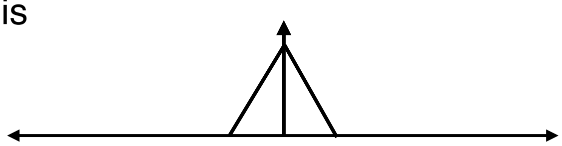


How do we get back our original signal?

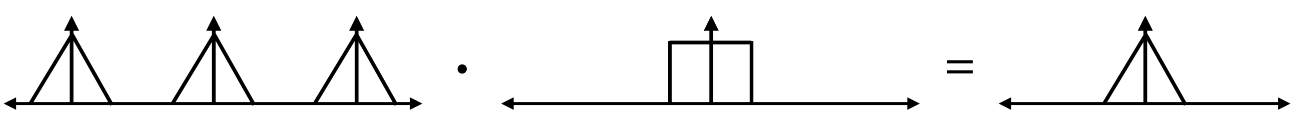
We need to multiply the spectrum by the following square wave:



The result is



which is the spectrum of our original function!



$\mathcal{F}\{f\,\text{Ш}_\Delta\}$ $\quad\cdot\quad$ $\Delta\Pi\,(k\Delta)$ $\quad=\quad$ $\mathcal{F}\{f\}$

$$\mathcal{F}\{f_s\} = \mathcal{F}\{f\,\text{Ш}_\Delta\}\,\Delta(k\,\Delta)$$

$$\Rightarrow f = \mathcal{F}^{-1}\left\{\mathcal{F}\{f_s \text{Ш}_\Delta\}\,\Delta\Pi(k\Delta)\right\} = \mathcal{F}^{-1}\left\{\mathcal{F}\{f_s \text{Ш}_\Delta\}\right\} * \mathcal{F}^{-1}\left\{\Pi(k/\Delta)\right\}$$

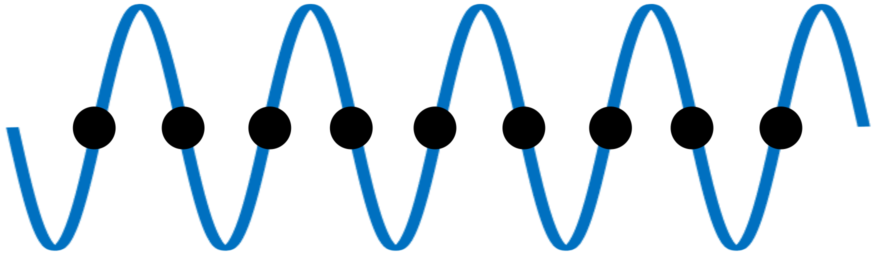
$$= f_s \text{Ш}_\Delta * \mathcal{F}^{-1}\left\{\Delta\Pi(k\Delta)\right\} = f_s \text{Ш}_\Delta * \operatorname{sinc}(x/\Delta).$$

$$f(x) = \sum_{n=-\infty}^{\infty} \boldsymbol{f}_n \operatorname{sinc}\left(\frac{x - n\Delta}{\Delta}\right)$$

Now we know that we can reconstruct a band-limited function from an infinite number of evenly spaced samples as long as the sampling is greater than twice the bandwidth.

What if the sampling is equal to twice the bandwidth? Can we still reconstruct the function?

Consider a sine of frequency *f*.   We sample at 2*f*.



When we look at the samples, we see that it could be a sine, or it could be the constant 0 function.

So sampling at exactly 2*f* is not enough.

Nyquist Sampling Theorem

Let *f* be a band-limited function with band-limit *B*. Then *f* is uniquely determined from its samples as long as the sampling frequency was greater than 2*B* according to the following expression:

$$f = f_s \text{Ш}_\Delta * \operatorname{sinc}(x/\Delta)$$

$$f(x) = \sum_{n=-\infty}^{\infty} \boldsymbol{f}_n \operatorname{sinc}\left(\frac{x - n\Delta}{\Delta}\right)$$

Outline

Sensors and Sampling

The Discrete Fourier Transform

Characterizing Errors

Forward Transform (Analysis Equations):

$$F(k) = \mathcal{F}\{f\}(k) = \int_{-\infty}^{\infty} f(x) \exp\left(-i\, 2\pi\, kx\right) dx$$

We can approximate this integral with a Riemann sum:

$$F(k) \approx \sum_j f(x_j) \exp\left(-i 2\pi k\, x_j\right) \Delta x_j$$

Let us assume that a finite number of samples $\{f(x_j)\}$ were gathered with a uniform spacing of $\Delta x_j = 1$.

$$F(k) \approx \sum_{n=0}^{N-1} f(n) \exp\left(-i 2\pi k\, n\right)$$

$$F(k) \approx \sum_{n=0}^{N-1} f(n) \exp\left(-i 2\pi k\, n\right)$$

Let $\boldsymbol{f}$ be the vector $(f(0), f(1), \ldots, f(N-1))$.

$$F(k) \approx \sum_{n=0}^{N-1} \boldsymbol{f}_n \exp\left(-i 2\pi k\, n\right)$$

Let us consider the frequencies $\boldsymbol{k} = \left(0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N}\right)$.

$$F(\boldsymbol{k}_m) \approx \sum_{n=0}^{N-1} \boldsymbol{f}_n \exp\left(-i 2\pi \frac{m\, n}{N}\right)$$

Discrete Fourier Transform

Analysis Equations:

$$\mathrm{DFT}\{\boldsymbol{f}\}_m = \sum_{n=0}^{N-1} \boldsymbol{f}_n \exp\left(-i2\pi \frac{m\,n}{N}\right)$$

We have just shown that $\mathrm{DFT}\{\boldsymbol{f}\} \approx F(\boldsymbol{k})$.

Inverse Fourier Transform (Synthesis Equations):

$$f(x) = \mathcal{F}^{-1}\{F\}(x) = \int_{-\infty}^{\infty} F(k) \exp\left(i\,2\pi\, kx\right) dk$$

We can approximate this as a Riemann sum:

$$f(x) \approx \sum_j F(k_j)\, \exp\left(i\,2\pi k_j x\right)\, \Delta k_j$$

Considering $\boldsymbol{x} = (0, 1, \ldots, N-1)$ and $\boldsymbol{F} = \left(F(0), F\left(\frac{1}{N}\right), \ldots, F\left(\frac{N-1}{N}\right)\right)$

$$f(\boldsymbol{x})_n \approx \frac{1}{N} \sum_{m=0}^{N-1} \boldsymbol{F}_m\, \exp\left(i2\pi \frac{m\,n}{N}\right)$$

Inverse Discrete Fourier Transform

Synthesis Equations:

$$\mathrm{IDFT}\{\boldsymbol{F}\}_n = \frac{1}{N}\sum_{m=0}^{N-1} \boldsymbol{F}_m \, \exp\left(i2\pi\frac{m\,n}{N}\right)$$

We have just shown that $\mathrm{IDFT}\{\boldsymbol{F}\} \approx f(\boldsymbol{x})$.

Discrete Fourier Transform

Analysis Equations:

$$\mathrm{DFT}\{\boldsymbol{f}\}_m = \sum_{n=0}^{N-1} \boldsymbol{f}_n \exp\left(-i2\pi\frac{m\,n}{N}\right)$$

Synthesis Equations:

$$\mathrm{IDFT}\{\boldsymbol{F}\}_n = \frac{1}{N}\sum_{m=0}^{N-1} \boldsymbol{F}_m \, \exp\left(i2\pi\frac{m\,n}{N}\right)$$

Note: both the DFT and the IDFT can be done by a computer.

The Discrete Fourier Transform and the Inverse Discrete Fourier Transform are both square matrix multiplications.

The computational complexity of square matrix-vector multiplication is $\mathbb{O}(n^2)$.

There is a faster way to implement the DFT (and the inverse DFT) called the Fast Fourier Transform (FFT) algorithm.

The computational complexity of the FFT is $\mathbb{O}(n \log n)$.

Outline

Sensors and Sampling

The Discrete Fourier Transform

Characterizing Errors

For simplicity, we will suppose that we gathered N samples:

$$\boldsymbol{f} = (f(0), f(1), \ldots, f(N-1))\,.$$

Consider the distribution $f\,\text{Ш}\,\Pi_D$.

$$\mathcal{F}\left\{f_s\,\text{Ш}\,\Pi_D\right\} = \int_{-\infty}^{\infty} f(x)\,\text{Ш}(x)\,\Pi_D(x)\exp\left(-i\,2\pi\,kx\right)dx$$

$$= \int_{-D/2}^{D/2} f(x)\,\text{Ш}(x)\exp\left(-i\,2\pi\,kx\right)dx$$

$$\mathcal{F}\left\{f\,\text{Ш}\,\Pi_D\right\}(k) = \int_{-D/2}^{D/2} f(x)\,\text{Ш}(x)\exp\left(-i\,2\pi\,kx\right)dx$$

$$= \sum_{n=^-N/2}^{N/2} f(n)\exp\left(-i\,2\pi\,kx\right) \;= \sum_{n=0}^{N-1} \boldsymbol{f}_n \exp\left(-i\,2\pi\,kn\right)$$

where we have assumed that the unknown values of *f* are related to ***f*** by its periodic extension, defined as $\boldsymbol{f}_{p.e.}(n) = \boldsymbol{f}(n \bmod N)$.

When we evaluate the above expression at $k = m/N$, we get

$$\mathcal{F}\{f\,\text{Ш}\,\Pi_D\}\,(m/N) = \sum_{n=0}^{N-1} \boldsymbol{f}_n\,\exp\left(-i\,2\pi\frac{mn}{N}\right)$$

This is just the DFT of $\boldsymbol{f}$.

Let's consider, again, the Fourier Transform of $f\, Ш\, \Pi_D$.

$$\mathcal{F}\left\{f\, Ш\, \Pi_D\right\}(k) = (F * Ш * b)(k)$$

where $b(k) = D\,\mathrm{sinc}(k\,D)$.

Equating the two expressions yields

$$\boxed{\mathrm{DFT}\{\boldsymbol{f}\}_m = (F * Ш * b)\left(\frac{m}{N}\right).}$$

So we see that the DFT attains samples of F corrupted by a convolution with a comb function (called aliasing) and a sinc (called blurring).

Summary

Nyquist Theorem

We can approximate the Fourier Transform with a computer using the Discrete Fourier Transform.

The DFT approximations are corrupted by aliasing and blurring.