Representations Tutorial

Jason Ritt
jason_ritt@brown.edu
Scientific Director of Quantitative Neuroscience

ROBERT J. & NANCY D. CARNEY
INSTITUTE FOR BRAIN SCIENCE
BROWN UNIVERSITY

Carney Institute for Brain Science, August 23rd, 2022

Note: Presentation modified from original to comply with data confidentiality agreements
and to adapt for a general audience

Expectations for today

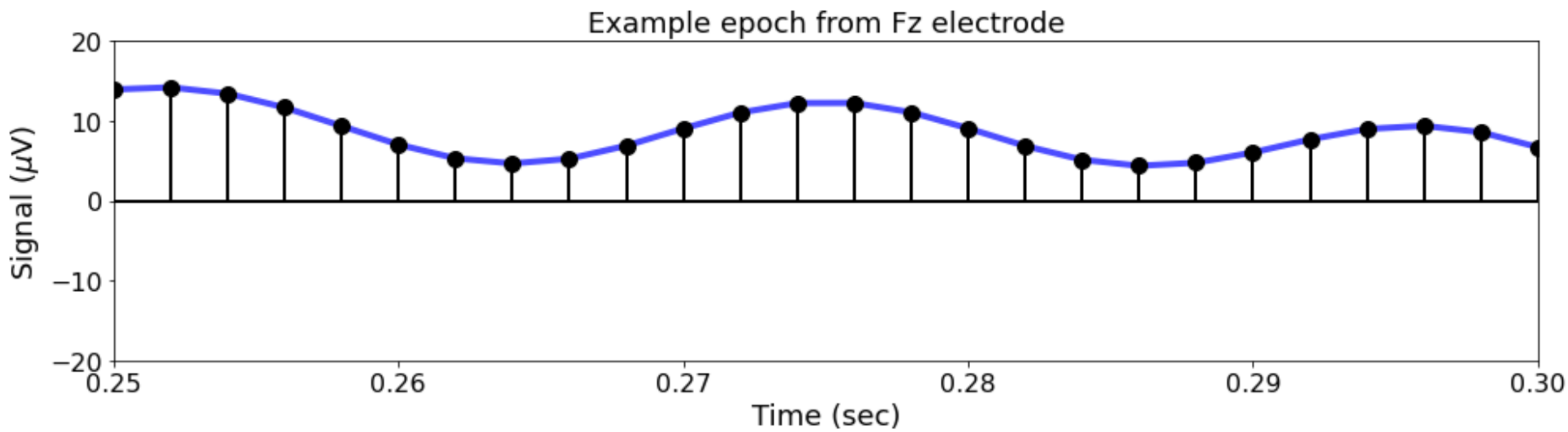
The unifying idea behind all the methods we will discuss is to ***represent a given time series as a linear combination of simpler and/or more interpretable components***

Time-frequency analysis and multi-channel transformations are *vast* subjects, with many, many approaches and algorithms, and ways of describing them

Participant backgrounds are diverse across experimental, computational, and mathematical experience

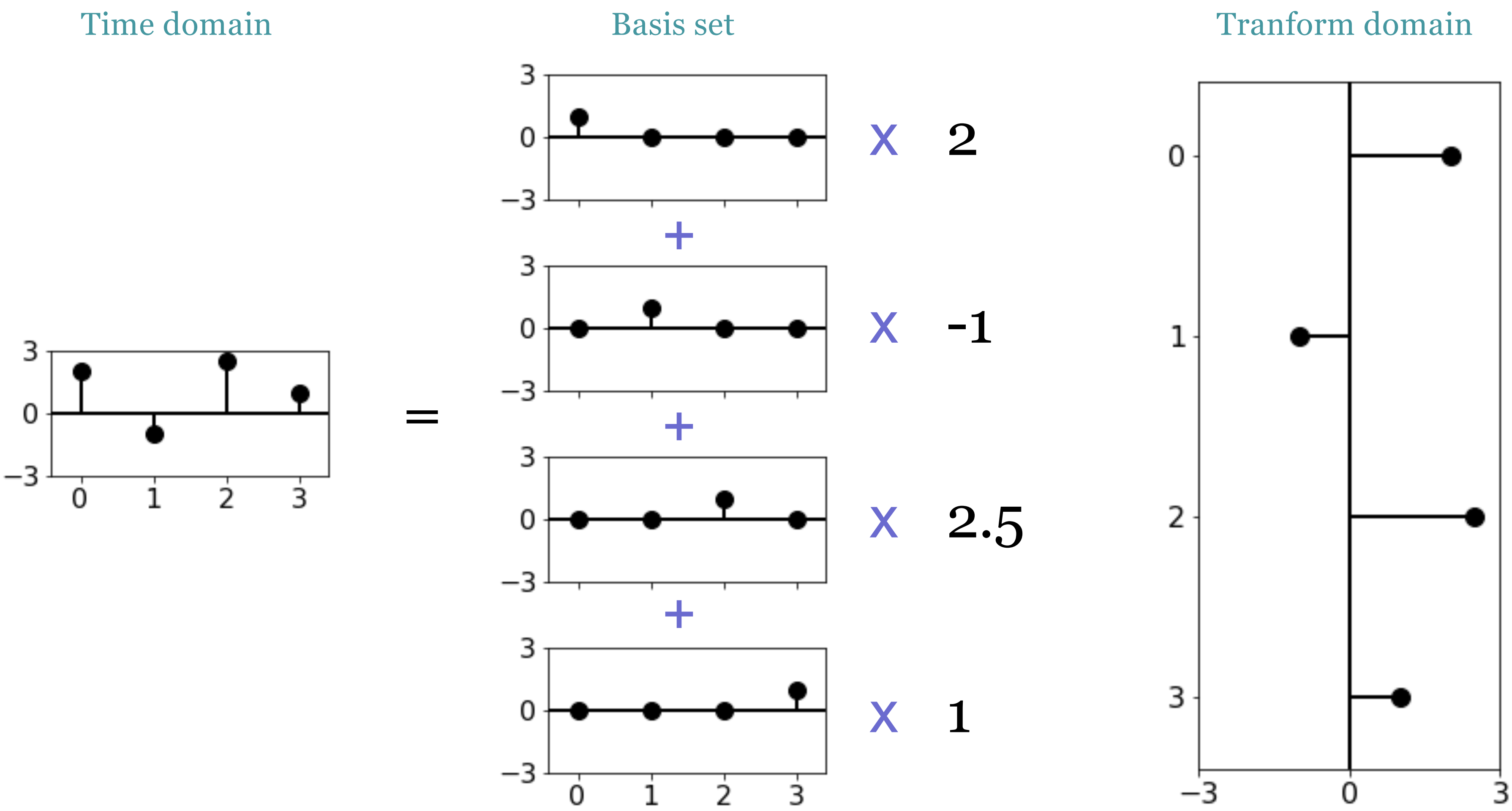
This tutorial is meant to connect core ideas to “minimal” code examples; some methods will be expanded in later tutorials, and some may be starting points for challenge work next week

We think in continuous time, but computers work in discrete time

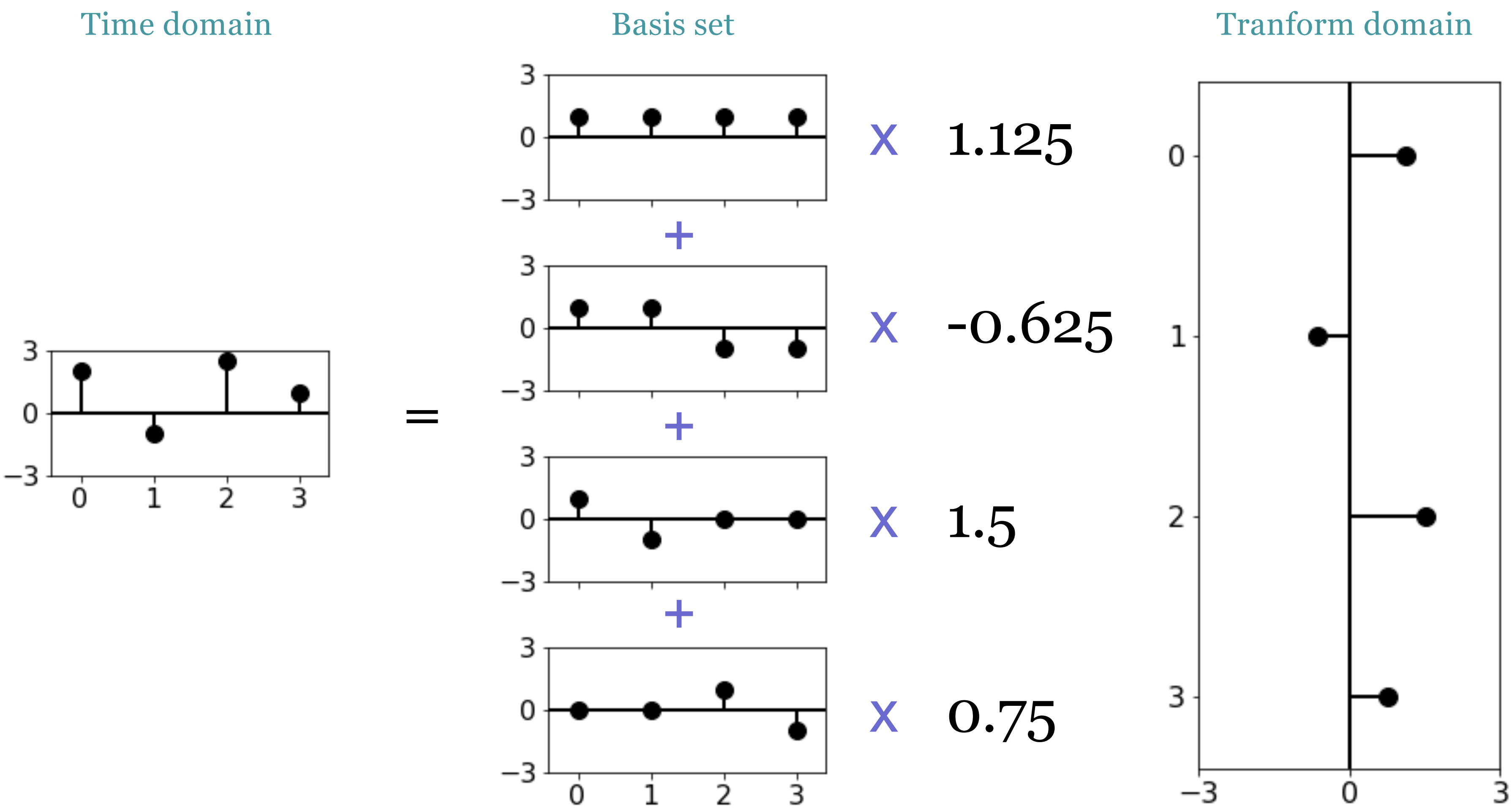


Many transform methods are intuitive when applied to only a few samples, but appear complicated at the large scales of sampled EEG data

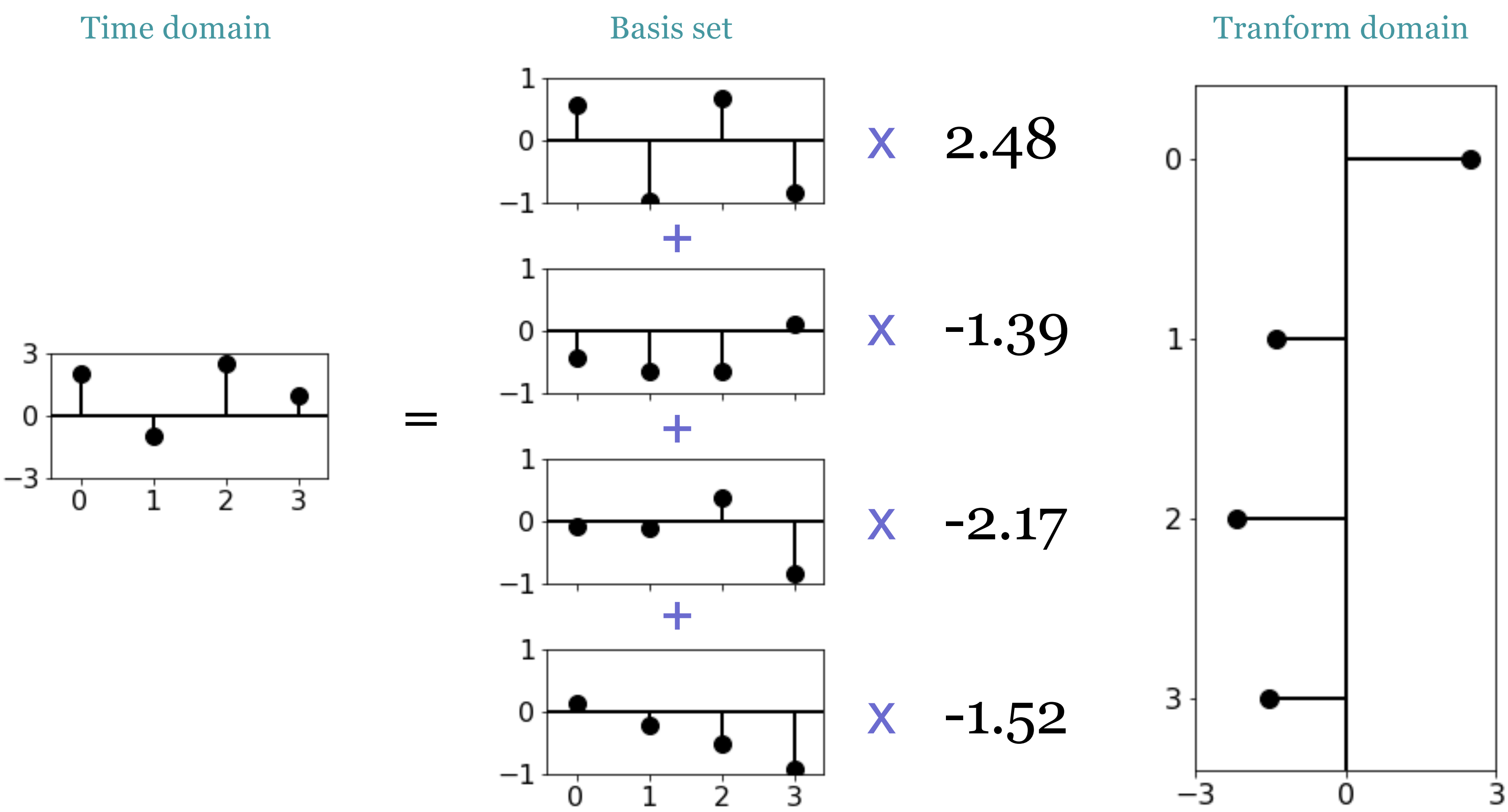
Representation: Natural basis (pure time basis)



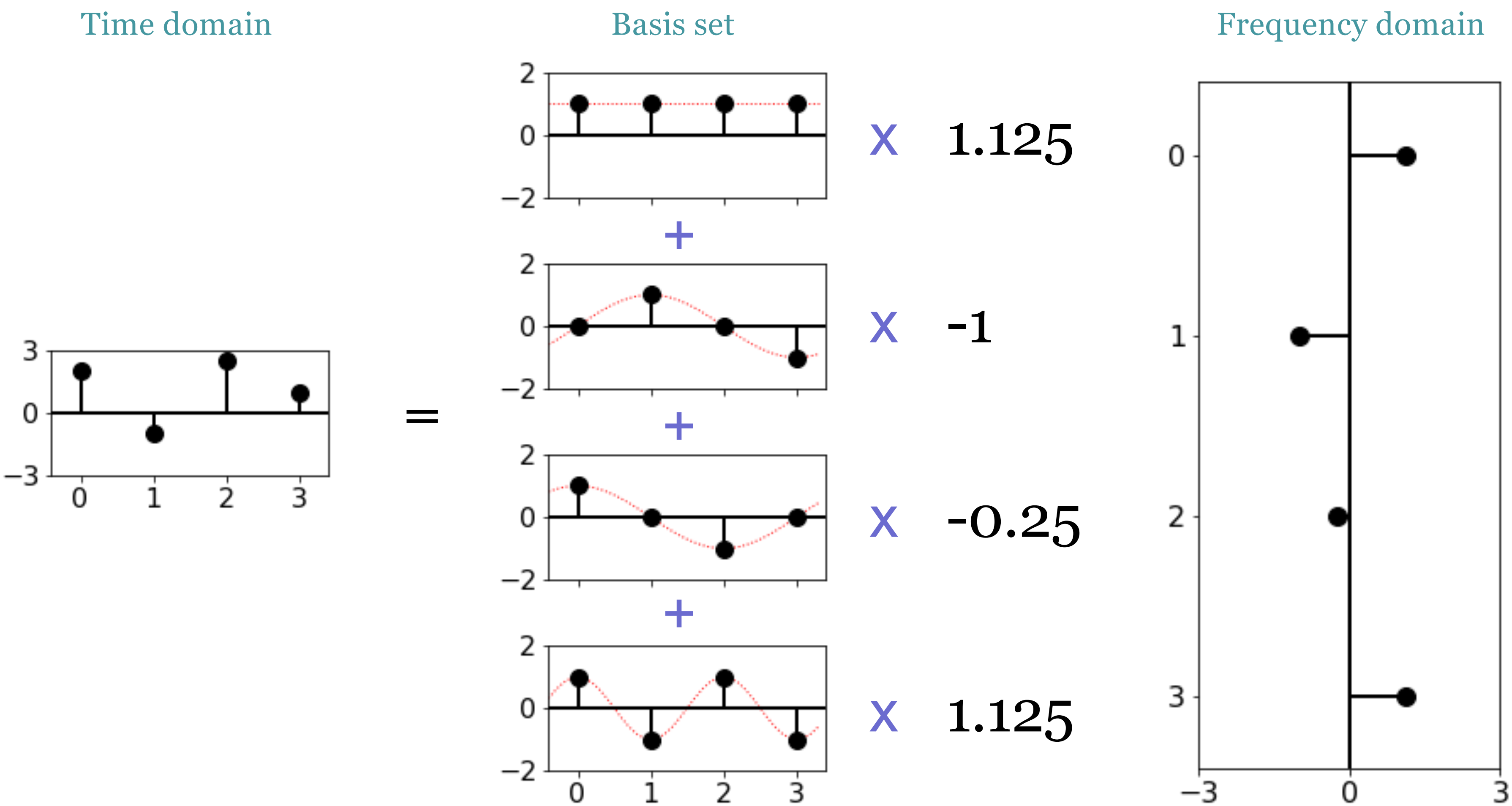
Representation: Haar-like basis (averages and differences)



Representation: random basis !



Representation: Fourier basis (sinusoids)



Fourier transforms

The **Fourier transform** of a signal is the list of coefficients for a linear combination of sinusoids with “harmonic” frequencies that produces that signal.

There are many “varieties” of Fourier transforms, that differ only in:

(1) the way the basis functions are organized, and

(2) what is assumed about the signal (is it periodic? finite length?)

There are also some technical differences between continuous time and discrete time cases.

The Fast Fourier Transform (FFT) is just a computational algorithm.

I will take the attitude that we want to think in continuous time, and treat the algorithms as a discrete approximation.

Some fundamental facts about Fourier transforms

The Fourier **frequency resolution** is set by the slowest sinusoid that completes one cycle within the recording duration. All other sinusoids are at integer multiples of this frequency.

Record for T seconds, your resolution is $df = 1/T$ Hz

For sampled data, the frequency range extends up to the sampling rate F_s Hz. However, frequencies about the **Nyquist rate** $F_s/2$ Hz are "aliased" to lower frequencies and do not provide additional information.

Record at F_s Hz, the **Fourier frequencies** are 0, df, $2df$, ... , $[F_s/2]$

There are **always as many FFT coefficients as original data points**.

All transform methods are constrained by the **uncertainty principle**, which says temporal and frequency resolution cannot both be high simultaneously.

Why power? The Pythagorean formula rules all

We usually consider **power spectral densities**, which are the squared magnitudes of the Fourier coefficients, rather than the **amplitude spectrum** of the magnitudes themselves.

The integral of the power spectrum equals the variance of the signal, which is by definition its power (**Parseval's relation**). From this point of view, what the Fourier transform does is *break the total variance down into variation at different frequency components*.

This is similar to the fact that for two independent random variables, their *variances* add, *not* their standard deviations.

Fundamentally, these relations are the Pythagorean formula applied to "function space" vectors.

There are many elaborations and variations…

The companion Jupyter notebook contains examples of the following:

Time-frequency analysis performs frequency transformations in multiple time “windows” spread across the recording duration.

Wavelet analysis is a variation that couples the window size to the frequency.

Multi-channel analyses include coherence, phase coupling, amplitude correlation, etc, all of which compute transforms of individual channels, and then compare parts of those transforms

Principal components analysis decides on basis functions by first looking at the data; components are chosen to “rank order” variance in different directions of the multi-channel set.