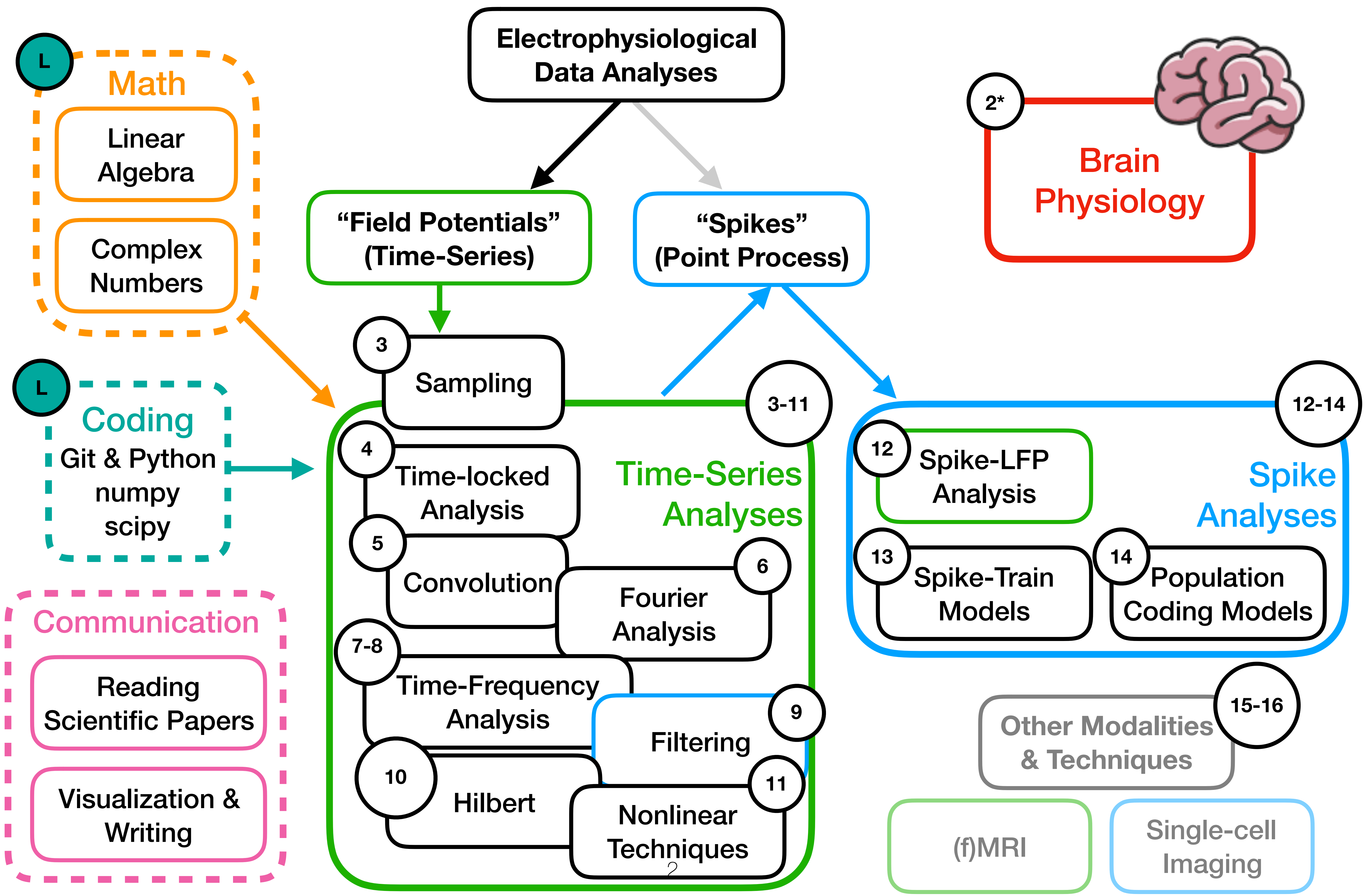
Filtering

Lecture 9

July 17, 2019

Course Outline: Road Map



Goals for Today

1. Understand the concept of filtering and convolution

2. Evaluate filter parameter choices

3. scipy.signal.filtwin tutorial

🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻

Why Filter?

Digital filters act, well, like a filter: it lets some frequencies through, and blocks out unwanted frequencies.

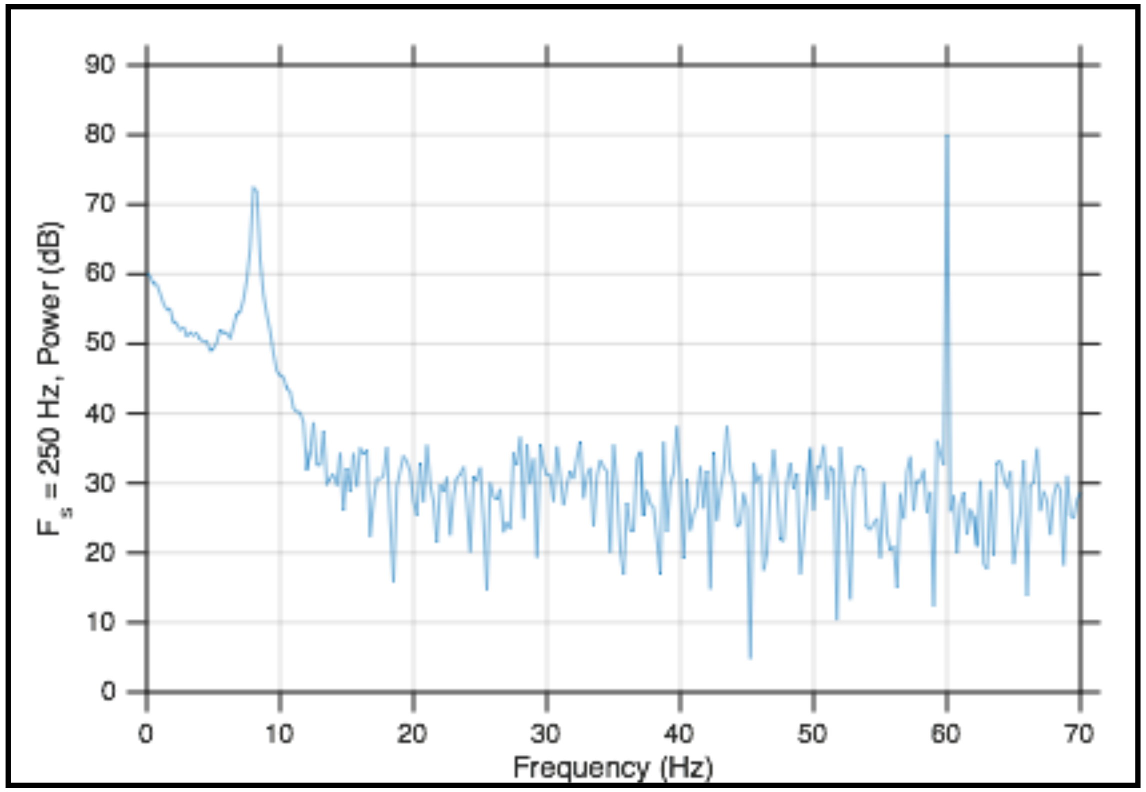
Much more intuitive to first think about in the frequency domain.

Come up with 3 examples of signals you can record, and which components you might want to filter out/keep from that signal.

Signal	Unwanted/Wanted Component	Frequencies
gopro audio during skydiving	hum + wind/ voice	
concert	audience/ music	
vocals/isolation	boys/girls	
bridges	resonant freqs	

How to Filter?

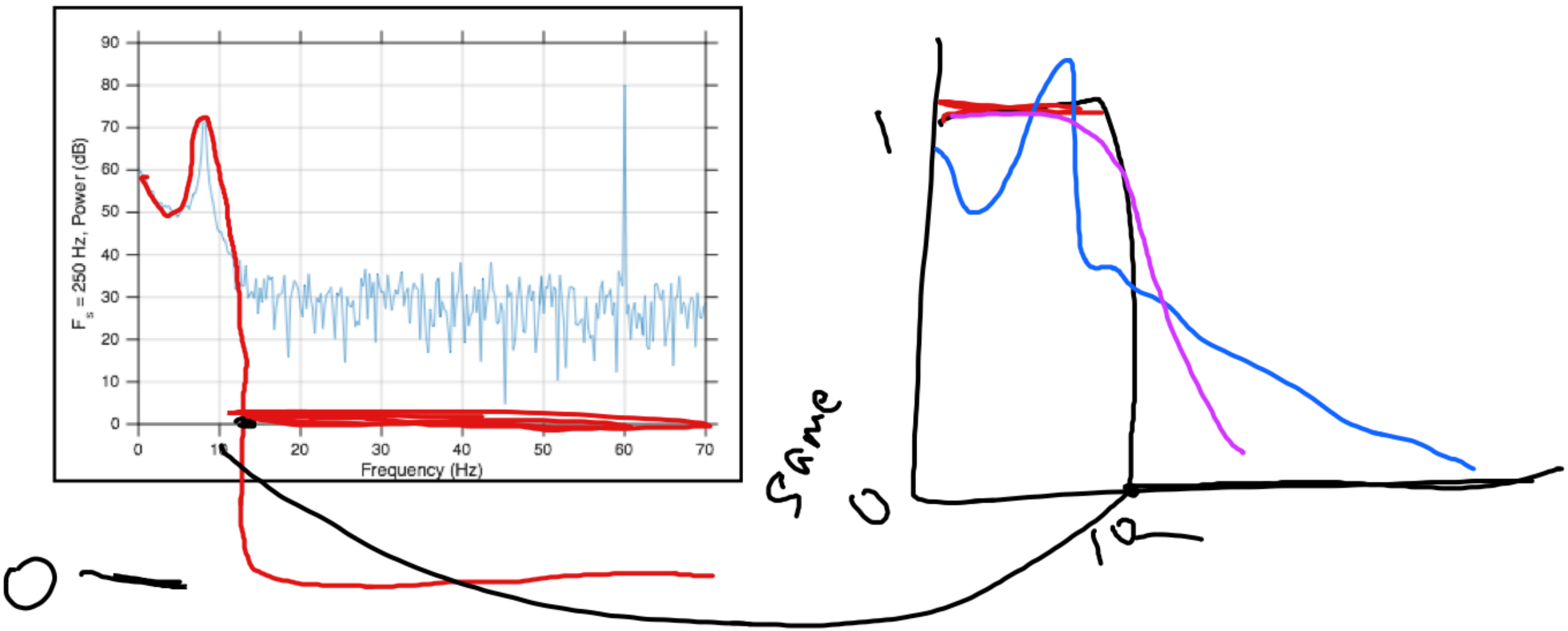
Given this signal you recorded, and assuming everything 10Hz and below is the true signal, draw the PSD of the ideal, noiseless signal.



If the only operation you're allowed is element-wise multiplication, draw what you would apply to achieve the ideal PSD.

Given this signal you recorded, and assuming everything 10Hz and below is the true signal, draw the PSD of the ideal, noiseless signal.

If the only operation you're allowed is element-wise multiplication, draw what you would apply to achieve the ideal PSD.



Frequency Response

$$X(f)H(f) = Y(f)$$

X(f): signal you record

Y(f): signal you want

H(f): filter frequency response

Note: this is multiplication of two vectors of complex numbers!

$$\boxed{\begin{aligned} x(t) \circledast h(t) &= y(t) \\ X(f)H(f) &= Y(f) \end{aligned}}$$

How to Filter

(Circular) Convolution Theorem

Convolution in time domain is equivalent to multiplication in frequency domain, and the converse is true as well (duality).

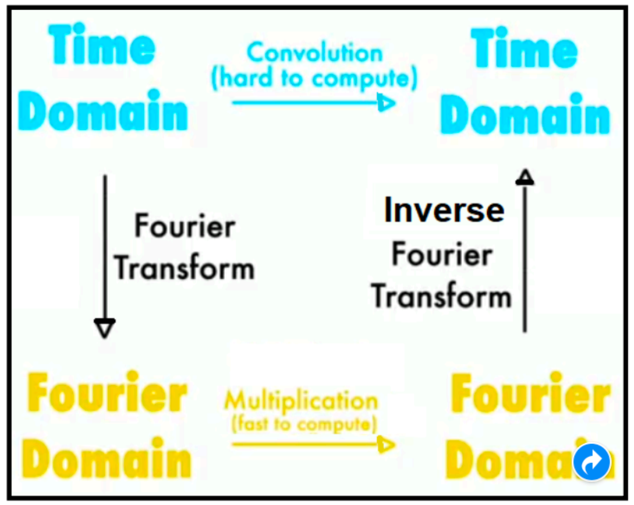
x: input
h: system's IRF
y: output

$$\boxed{\begin{aligned} x(t) \circledast h(t) &= y(t) \\ X(f)H(f) &= Y(f) \end{aligned}}$$

IRF: [2,-1,-1,0]

Input: x(n) = [1,0,3,7,2,5]

	n =	0	1	2	3	4	5	6	7	8	9	10
	1	2	-1	-1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0						
	3	0	0	6	-3	-3	0					
deltas	**7**	0	0	0	14	-7	-7	0				
	2	0	0	0	0	4	-2	-2	0			
	5	0	0	0	0	0	10	-5	-5	0	0	
		2	-1	5	11	-6	1	-7	-5	0	0	0



$$\boxed{\begin{aligned} x(t) \circledast h(t) &= y(t) \\ X(f)H(f) &= Y(f) \end{aligned}}$$

In python, we'll call **np.convolve()**, but it essentially performs FFT-multiplication-iFFT.

How to Filter

SciPy.org Sponsored By ENTHOUGHT

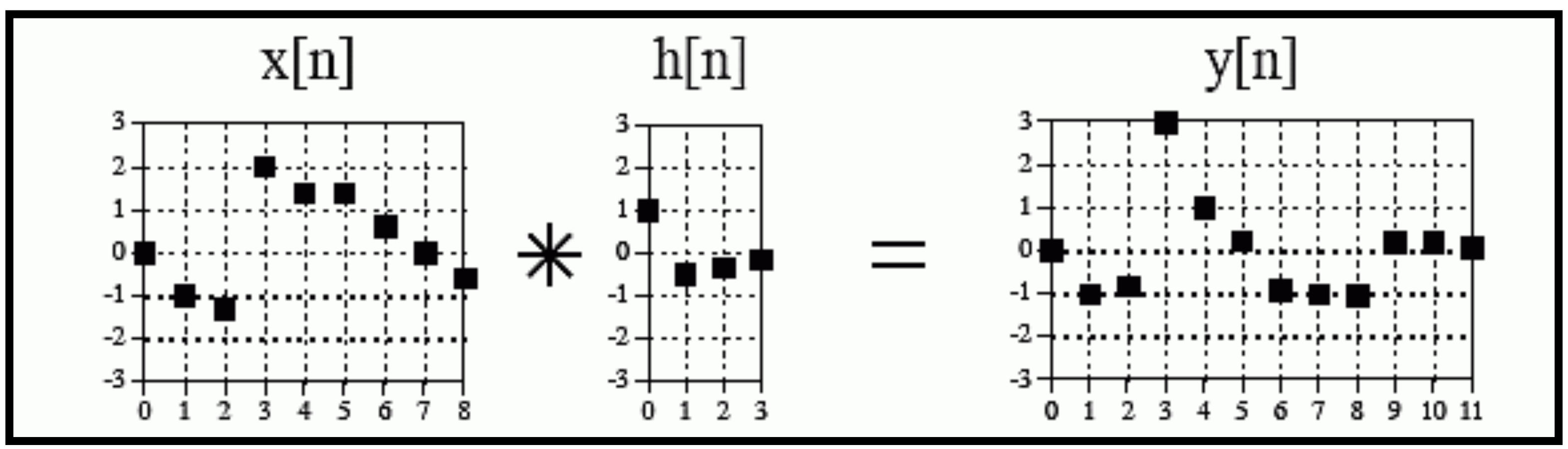
Scipy.org | Docs | NumPy v1.16 Manual | NumPy Reference | Routines | Mathematical functions

numpy.convolve

numpy.convolve(*a, v, mode='full'*) [source]

Returns the discrete, linear convolution of two one-dimensional sequences.

The convolution operator is often seen in signal processing, where it models the effect of a linear time-invariant system on a signal [1]. In probability theory, the sum of two independent random variables is distributed according to the convolution of their individual distributions.

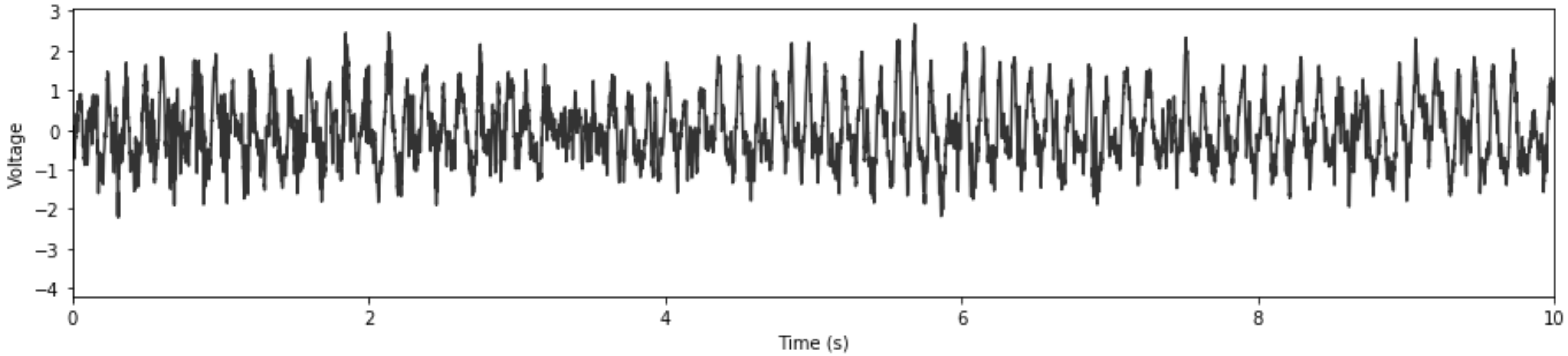


Signal Filter Filtered Signal

"Filter coefficients" is its **Impulse Response Function**

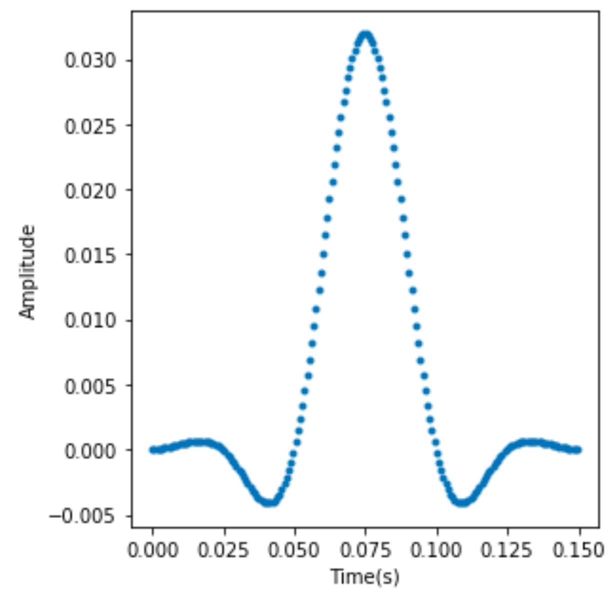
Impulse Response Function

Signal



convolve

Filter



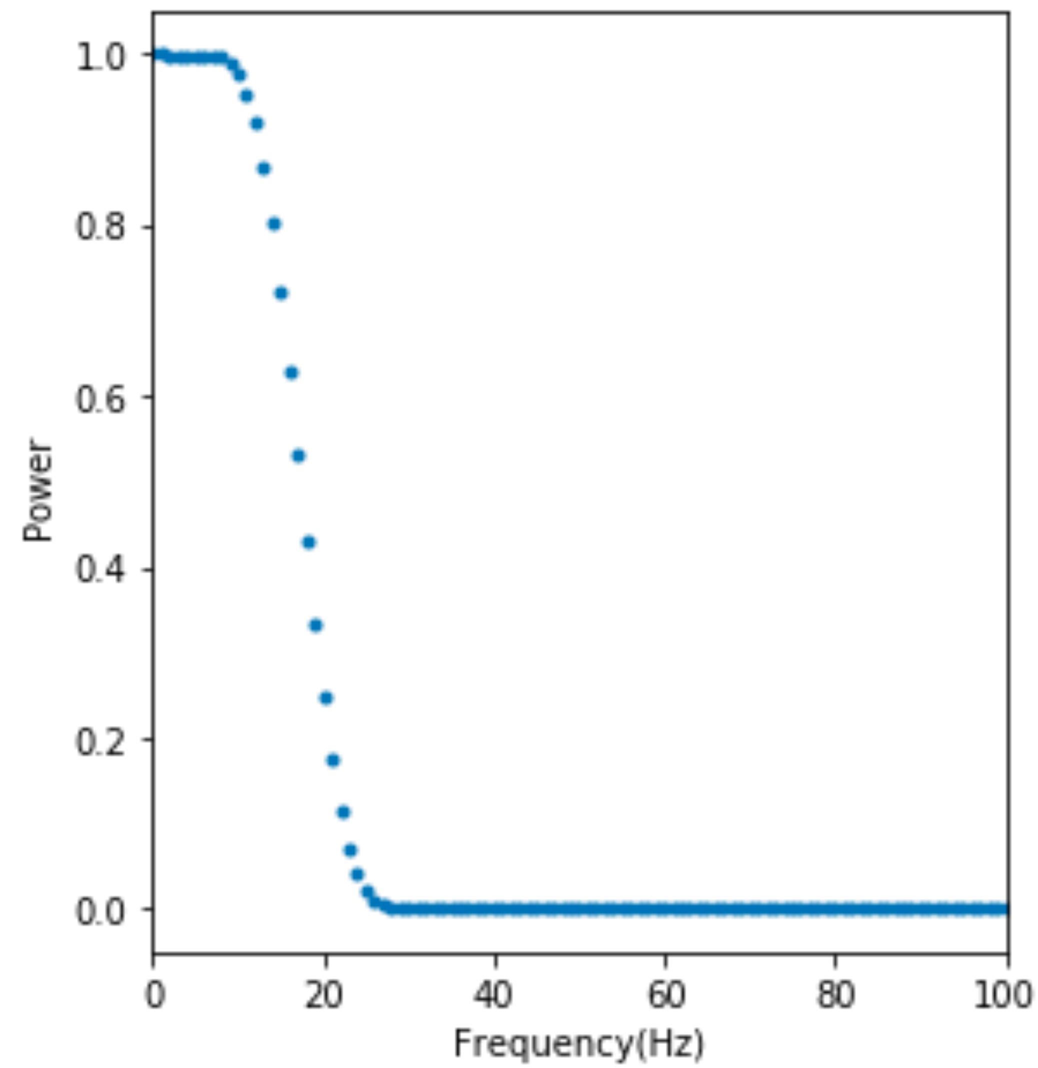
“Filter coefficients” is its **Impulse Response Function**

🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻🧠⚡💻

Magnitude & Phase Response

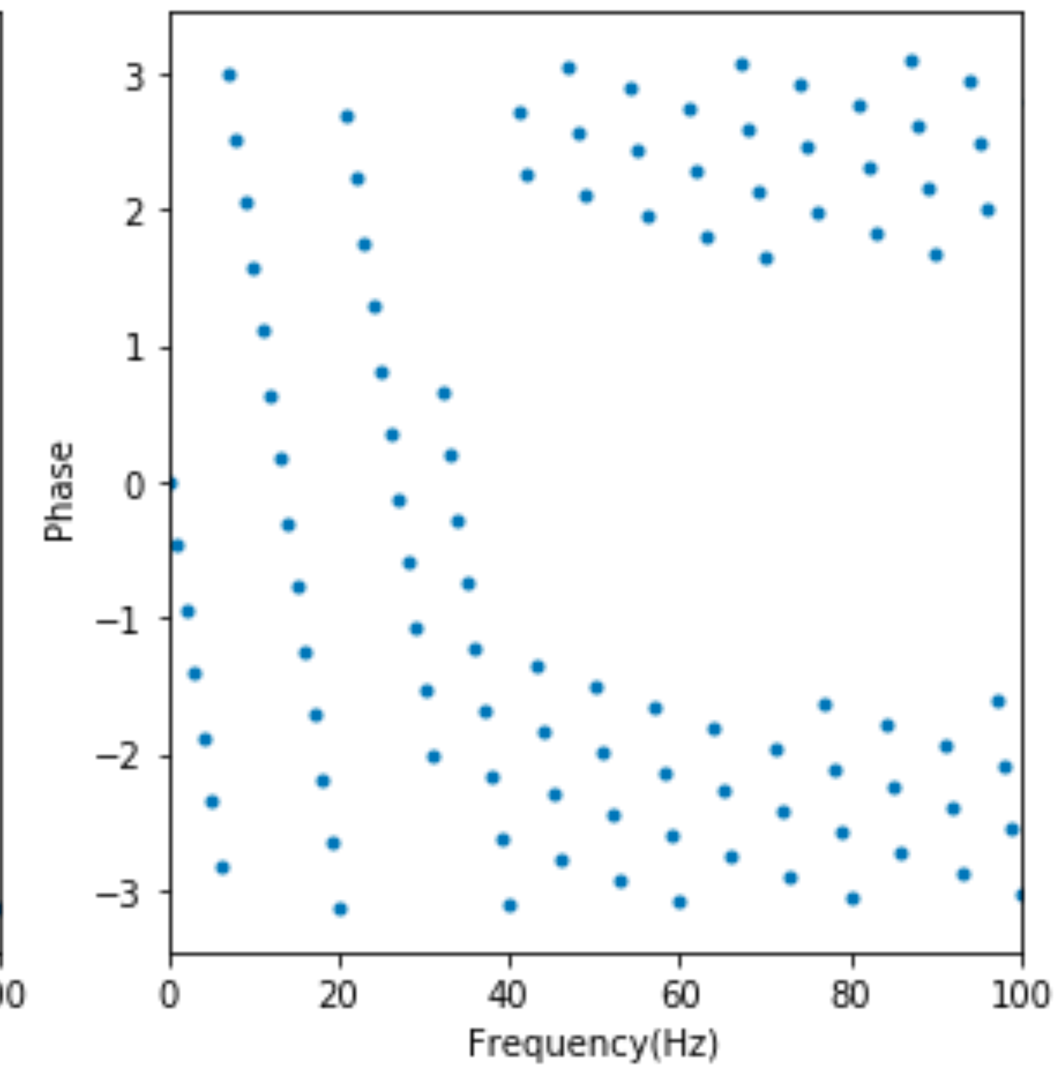
$$X(f)H(f) = Y(f)$$

Magnitude
Response (**gain**)



This is the thing we really care about.

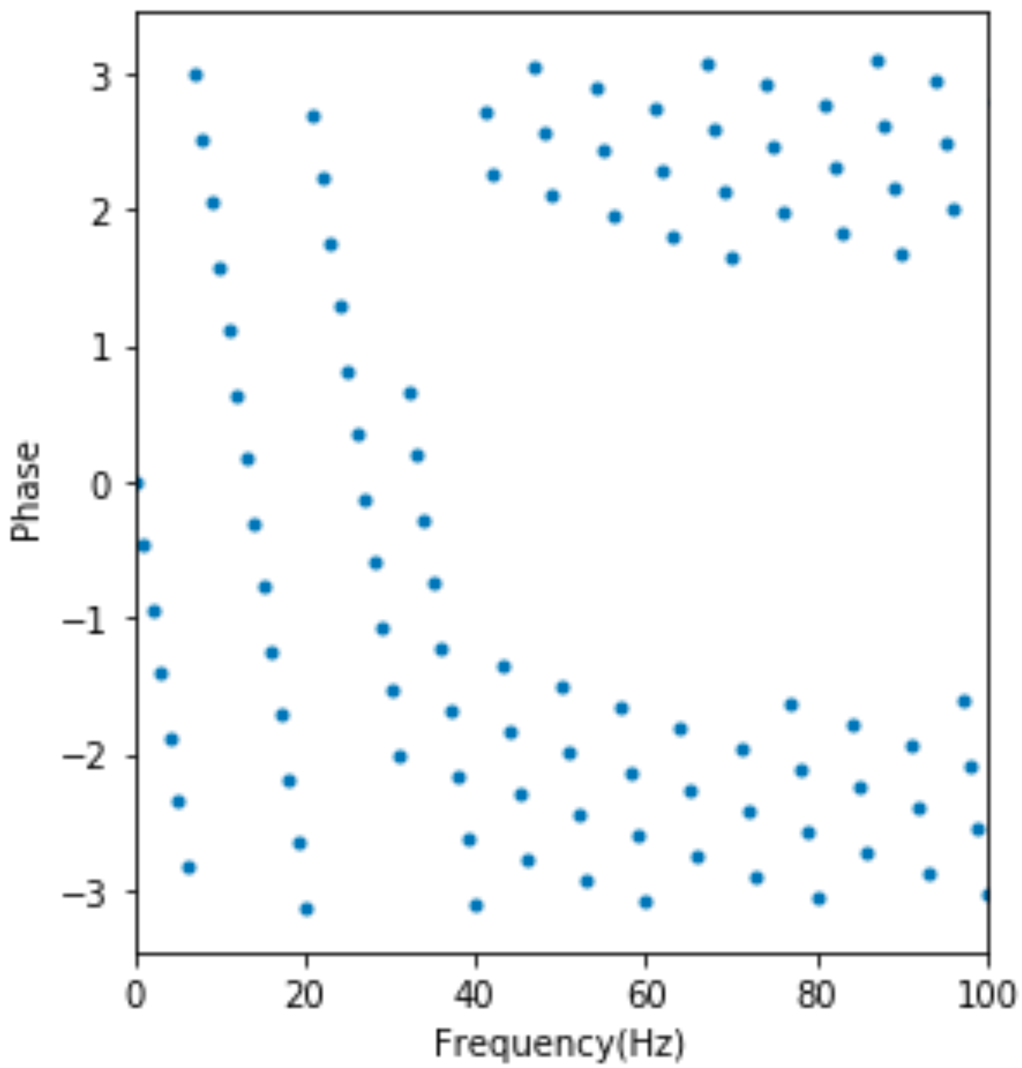
Phase
Response



This is the thing we want to be careful of.

Phase Response = Time Delay

Phase Response



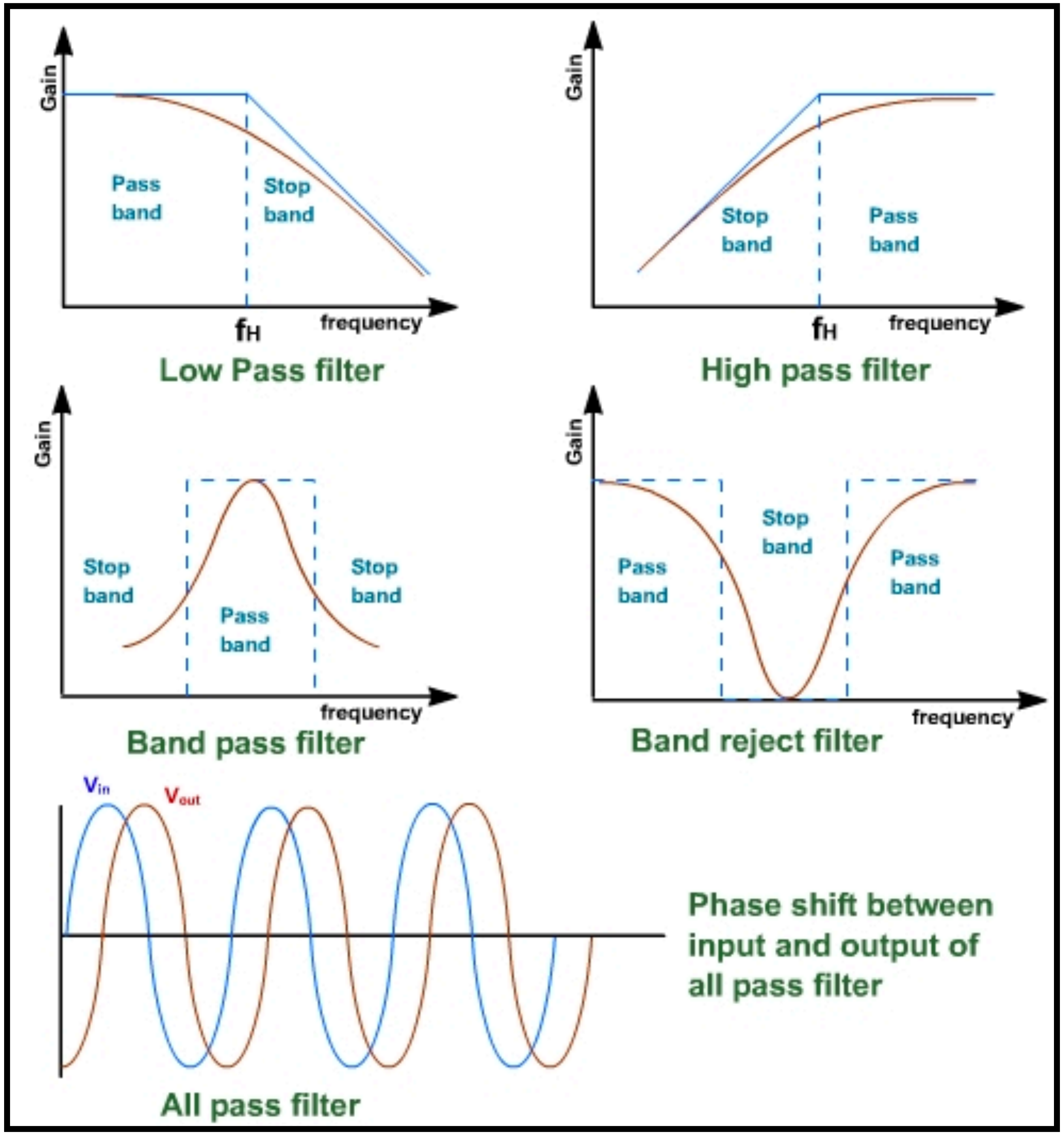
This is the thing we want to be careful of.

Non-zero phase response comes from time delay introduced by the causal filter.

(it's convolution, but shifted by a little bit)

Each data point, after filtering, is delayed by n/2 points, where n is the length of the filter.

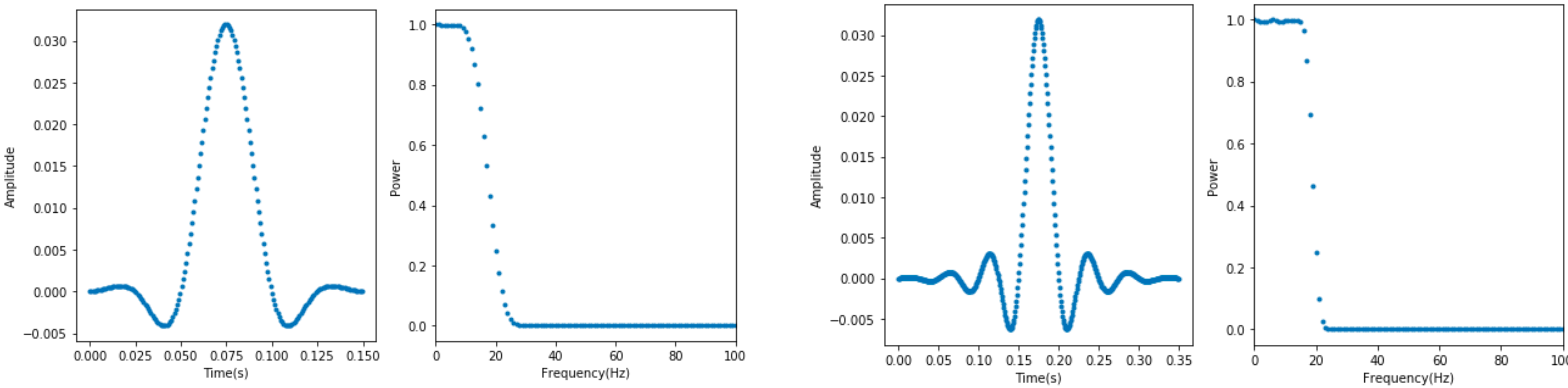
Magnitude Response - Types of Filter



3 min:
Come up with one example application for each type of filter.

Time-Frequency Resolution Tradeoff II

Since we're interested in the frequency response of the filter, the same concept applies:



Longer filter = better frequency resolution, but worse time resolution

Filter **roll-off** or **transition band**

Goals for Today

1. Understand the concept of filtering and convolution

2. Evaluate filter parameter choices

3. scipy.signal.filtwin tutorial

https://tinyurl.com/cogs118c-att