functional Magnetic Resonance Imaging – Methods

Denis Schluppeck

Visual Neuroscience Group
University of Nottingham, UK

2/4

Next ~~4~~ 3 lectures

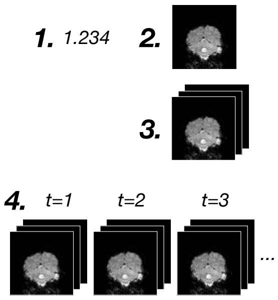
1. Spatial and temporal properties of fMRI
   (+ linearity, convolution)
2. Signal and Noise
   (+ Fourier domain, convolution)
3. Preprocessing of fMRI data
   (+ common software tools)
4. Statistics + experimental design
   (+ linear regression, GLM, multiple comparisons)

Matlab

?

Quick recap: data

1. numbers (=pixel/voxel)
2. a bunch of numbers on a grid (=slice)
3. a collection of slices (=volume)
4. many volumes over time, acquired every TR (=timeseries)



Data: indexing

- if we have a timeseries of volumes (in 3D + 1D = 4D), we need 4 “indices” or coordinates to uniquely identify a voxel (x,y,z,t)
- multi-dimensional arrays
- we can **slice** this data in different ways:

```
>> data(:,:,12,1) % get slice z=12 at t=1
>> data(32,:,:,1) % ??
>> data(1,1,12,:) % get timeseries at [1,1,12]
```


Data: indexing

- if we have a timeseries of volumes (in 3D + 1D = 4D), we need 4 “indices” or coordinates to uniquely identify a voxel (x,y,z,t)
- multi-dimensional arrays
- we can **slice** this data in different ways:

```
>> data(:,:,12,1) % get slice z=12 at t=1
>> data(32,:,:,1) % y/z slice at x=32, t=1
>> data(1,1,12,:) % get timeseries at [1,1,12]
```

Data: indexing

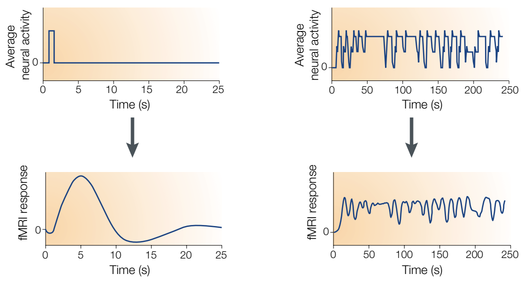
- if we have a timeseries of volumes (in 3D + 1D = 4D), we need 4 “indices” or coordinates to uniquely identify a voxel (x,y,z,t)
- multi-dimensional arrays
- we can **slice** this data in different ways:

```
>> data(:,:,12,1) % get slice z=12 at t=1
>> data(32,:,:,1) % y/z slice at x=32, t=1
>> data(1,1,12,:) % get timeseries at [1,1,12]
```


HRF

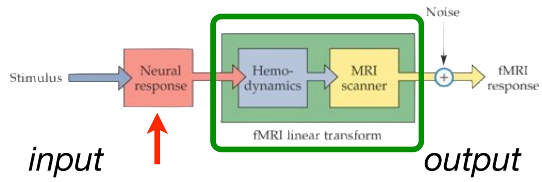
- the shape of the response to a brief impulse (e.g. visual stimulus) is called the haemodynamic response function (HRF)
- for a *linear system*, knowing the impulse response is sufficient to predict the response to an arbitrary input
- Linearity – clarification...
- Fourier domain / convolution
- Signal-to-noise / contrast-to-noise

impulse response *linear prediction*

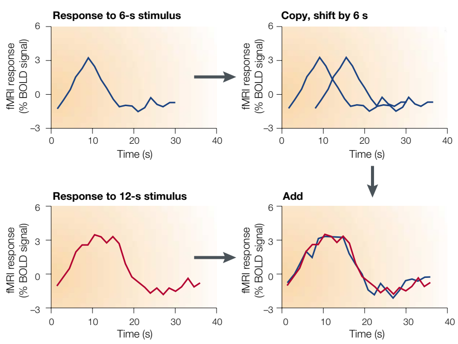


Heeger & Ress, NRN (2002)

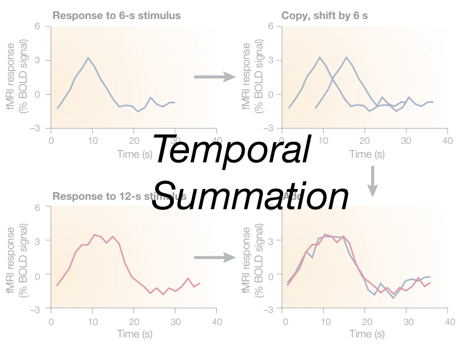
fMRI response as a linear system



Boynton et al (1996)



Heeger & Ress, NRN (2002)



Temporal Summation

Heeger & Ress, NRN (2002)

Neural activity: input to fMRI transform

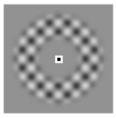
Neural activity: input to fMRI transform

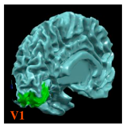
1. fMRI response is approximately a linear system

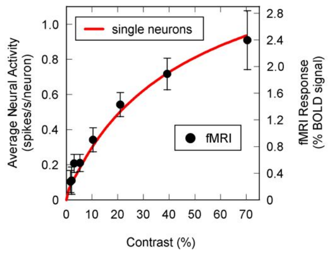
Neural activity: input to fMRI transform

1. fMRI response is approximately a linear system
2. neural activity is **not** a linear transform of e.g. visual stimulus
   - neuronal firing rates are > 0 (so at least half-rectifying)
   - response to visual contrast saturates (contrast response function)

fMRI response, firing rates







Heeger et al (2000) Nature Neurosci, 3:631+

Linearity does not always hold

Linearity does not always hold

1. very brief events (threshold)

Linearity does not always hold

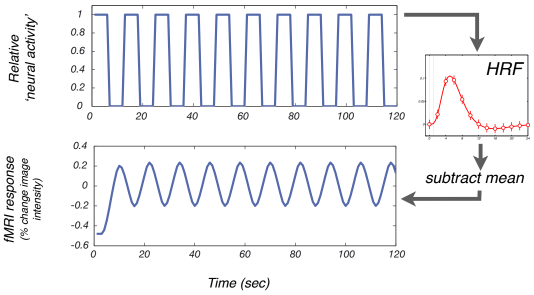
1. very brief events (threshold)
2. “refractory” effects for very closely spaced events

   cf. fMRI adaptation

Simulation
fMRI Response in a block design experiment

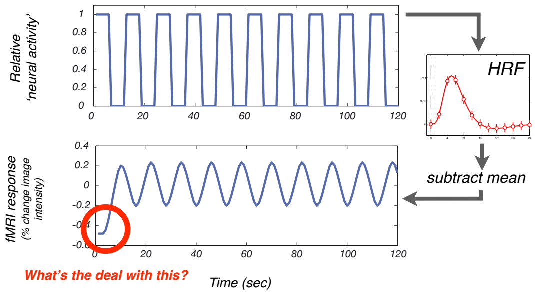
Block alternation

Stimulus alternation frequency = 1/12 Hz; (12s cycle)



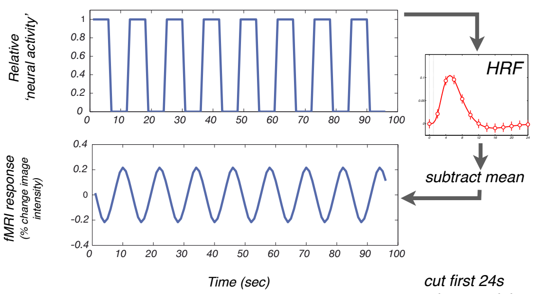
Block alternation

Stimulus alternation frequency = 1/12 Hz; (12s cycle)



Block alternation

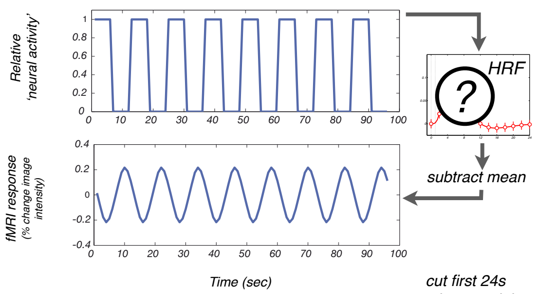
Stimulus alternation frequency = 1/12 Hz; (12s cycle)



cut first 24s
8 cycles remaining

Block alternation

Stimulus alternation frequency = 1/12 Hz; (12s cycle)



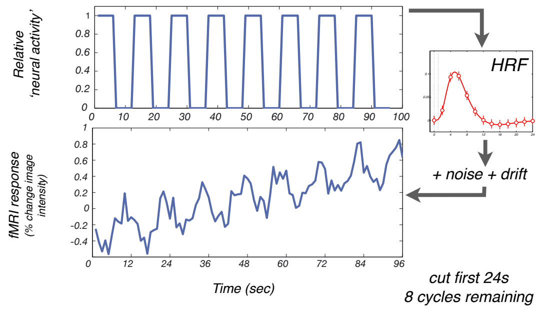
cut first 24s
8 cycles remaining

Noise

- measured data is never perfect...
- sources of (unwanted) variability: *heart beat, breathing, movements, ...*
- in fMRI data we usually (**high-frequency) 'noise'** and **drift**

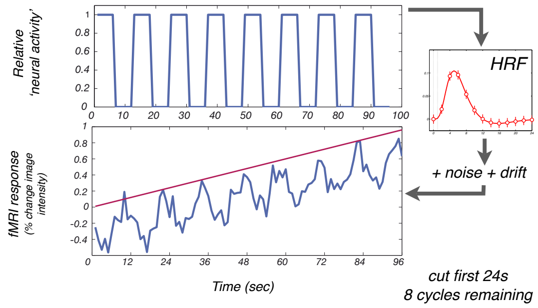
Block alternation / drift

Stimulus alternation frequency = 1/12 Hz; (12s cycle)



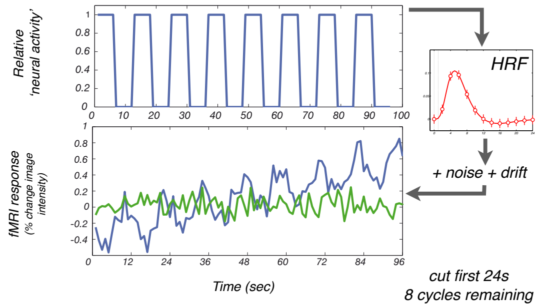
Block alternation / drift

Stimulus alternation frequency = 1/12 Hz; (12s cycle)



Block alternation / noise

Stimulus alternation frequency = 1/12 Hz; (12s cycle)



Time / Fourier Domain

Time domain versus Fourier domain

- compare to what you know about image domain ↔ *k*-space
- two different ways of looking at a signal: one in terms of time: ***s, ms***, the other in terms of frequencies: ***Hz (s^{-1}), cycles/scan***
- *Mathtools (Eero Simoncelli, NYU)* ***http://www.cns.nyu.edu/~eero/math-tools/*** *contains additional links to www / books*

T

Time / signal domain

x *'Fourier transform'* → y

$\mathcal{F}$

Frequency domain

x ← *'Inverse Fourier transform'* y

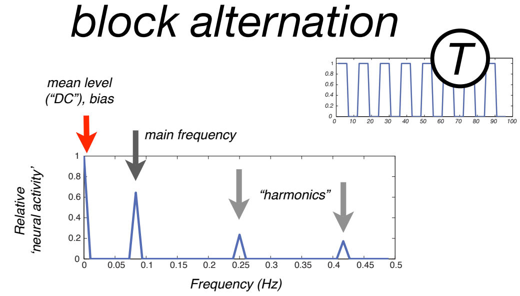
data

intensity as a function of time

complex numbers

frequency
phase

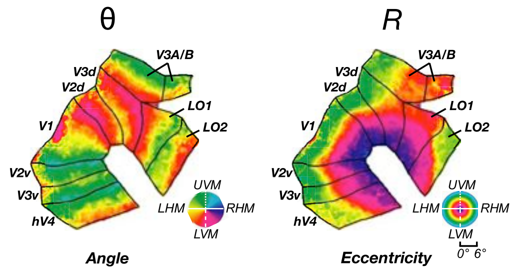
Fourier transform of block alternation



Where do the specific numbers come from?
1/12 s = 0.083Hz

Retinotopic / topographic Mapping

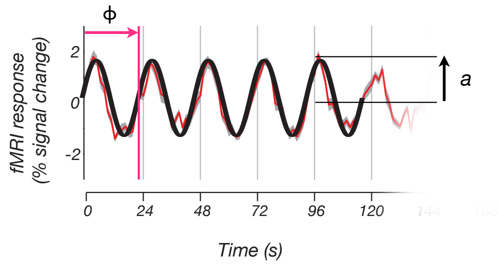
Lecture on "Vision" by Dr Peirce



Larsson & Heeger (2006) J Neurosci

Clever choice of stimulus: map "spatial location" into temporal delay (travelling wave of activity)

amplitude
coherence: [0, 1]
phase: $0 \leq \phi < 2\pi$



Lots of Fourier transforms...

	time domain	*$\mathcal{F}$ourier domain*
Fourier Transform	*continuous, infinite*	*continuous, infinite*
Fourier Series	*continuous, periodic*	*discrete, infinite*
DTFT	*discrete, infinite*	*continuous, periodic*
DFS	*discrete, periodic*	*discrete, periodic*
DFT	***discrete, finite***	***discrete, finite***

FFT Algorithm

- Computes DFT (discrete Fourier Transform) of finite length input
- Very efficient for inputs of lengths *$N = 2^n$*
- Produces 2 outputs, each of size/length equal to that of the input:
  real part (cosine coefficients)
  imaginary part (sine coefficients)

```
>> fftdemo     % matlab
```

$\mathcal{T}$ Time / signal domain

$\mathcal{F}$ Frequency domain

'Fourier transform'

x → y

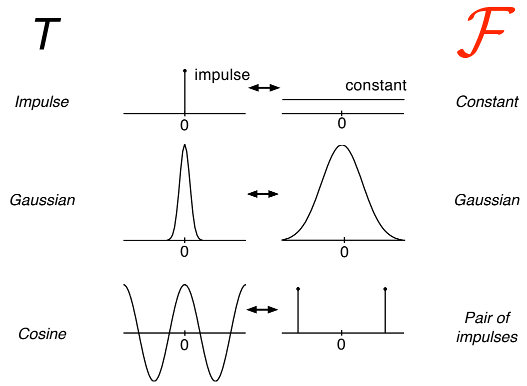
`fft(x)`

'Inverse Fourier transform'

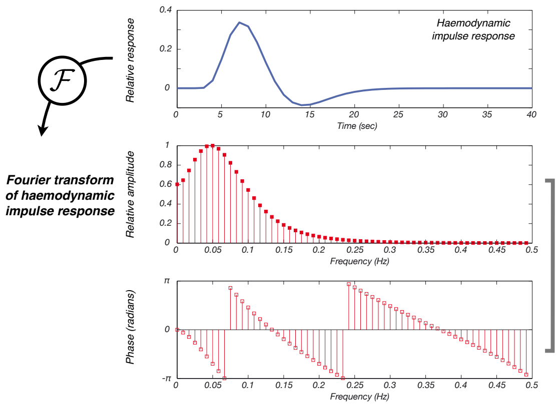
x ← y

`ifft(y)`

```
help fft, help ifftshift
help ifft, help abs
```

Eero Simoncelli, NYU



Convolution

Convolution

Discrete-time signal: $x[n] = [x_1, x_2, x_3, \ldots]$

A system or transform maps an input signal into an output signal:

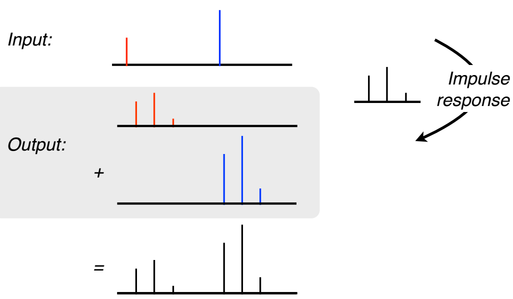
$$y[n] = T\{x[n]\}$$

A shift-invariant, linear system can always be expressed as a convolution:

$$y[n] = \sum x[m] \cdot h[n-m]$$

where h[n] is the impulse response.

Convolution as a sum of impulses



Convolution as sequence of weighted sums

past present future

0 0 0 1 0 0 0 0 0 input (impulse)

1/8 1/4 1/2 ⟶ weights

0 0 0 1/2 1/4 1/8 0 0 0 output (impulse response)

0 0 0 1 1 1 1 1 1 input (step)

1/8 1/4 1/2 ⟶ weights

0 0 0 1/2 3/4 7/8 7/8 7/8 7/8 output (step response)

☣ Convolution as matrix multiplication

$$\begin{bmatrix} \cdot \\ 5 \\ 2 \\ -3 \\ 4 \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & & & & & & \\ & 1 & 2 & 3 & 0 & 0 & 0 & & \\ & 0 & 1 & 2 & 3 & 0 & 0 & & \\ & 0 & 0 & 1 & 2 & 3 & 0 & & \\ & 0 & 0 & 0 & 1 & 2 & 3 & & \\ & & & & & \cdot & \cdot & \cdot & \\ & & & & & & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ 1 \\ 2 \\ 0 \\ 0 \\ -1 \\ \cdot \end{bmatrix}$$

Linear system ↔ matrix multiplication

Shift-invariant linear system ↔ 'Toeplitz' matrix

Matrix multiplication ??

A is a 2 by 2 matrix $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3.2 \end{bmatrix}$ *x is a vector (2 by 1 matrix)*

Matrix multiplication ??

A is a 2 by 2 matrix $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3.2 \end{bmatrix}$ *x is a vector (2 by 1 matrix)*

$$\begin{bmatrix} 1 \times 2.5 & +0 \times 3.2 \\ 0 \times 2.5 & +2 \times 3.2 \end{bmatrix}$$

Matrix multiplication ??

A is a 2 by 2 matrix $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3.2 \end{bmatrix}$ *x is a vector (2 by 1 matrix)*

$$\begin{bmatrix} 1 \times 2.5 & +0 \times 3.2 \\ 0 \times 2.5 & +2 \times 3.2 \end{bmatrix}$$

Matrix multiplication ??

A is a 2 by 2 matrix $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3.2 \end{bmatrix}$ *x is a vector (2 by 1 matrix)*

$$\begin{bmatrix} 1 \times 2.5 & +0 \times 3.2 \\ 0 \times 2.5 & +2 \times 3.2 \end{bmatrix}$$

Matrix multiplication ??

A is a 2 by 2 matrix $\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}2.5\\ 3.2\end{bmatrix}$ x is a vector (2 by 1 matrix)

$$\begin{bmatrix}1\times 2.5 & +0\times 3.2\\ 0\times 2.5 & +2\times 3.2\end{bmatrix}$$

Matrix multiplication ??

A is a 2 by 2 matrix $\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}2.5\\ 3.2\end{bmatrix}$ x is a vector (2 by 1 matrix)

$$\begin{bmatrix}1\times 2.5 & +0\times 3.2\\ 0\times 2.5 & +2\times 3.2\end{bmatrix}$$

Matrix multiplication ??

A is a 2 by 2 matrix $\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}2.5\\ 3.2\end{bmatrix}$ x is a vector (2 by 1 matrix)

$$\begin{bmatrix}1\\ 0\end{bmatrix}2.5+\begin{bmatrix}0\\ 2\end{bmatrix}3.2=\begin{bmatrix}2.5\\ 6.4\end{bmatrix}$$

weighted sum of columns ... $ax_1 + bx_2$ ... should ring a bell!

Convolution as matrix multiplication

$$\begin{bmatrix}\cdot\\ 5\\ 2\\ -3\\ 4\\ \cdot\\ \cdot\end{bmatrix}=\begin{bmatrix}\cdot & \cdot & \cdot & & & & & \\ & 1 & 2 & 3 & 0 & 0 & 0 & \\ & 0 & 1 & 2 & 3 & 0 & 0 & \\ & 0 & 0 & 1 & 2 & 3 & 0 & \\ & 0 & 0 & 0 & 1 & 2 & 3 & \\ & & & & & \cdot & \cdot & \cdot\\ & & & & & \cdot & \cdot & \cdot\end{bmatrix}\begin{bmatrix}\cdot\\ 1\\ 2\\ 0\\ 0\\ -1\\ \cdot\end{bmatrix}$$

Columns contain shifted copies of the impulse response.

Linear system ↔ matrix multiplication

Shift-invariant linear system ↔ 'Toeplitz' matrix

Convolution as matrix multiplication

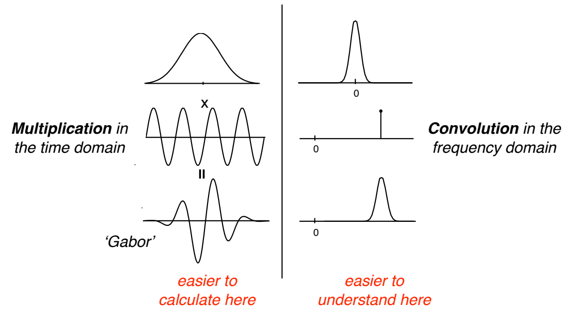
$$\begin{bmatrix}\cdot\\ 5\\ 2\\ -3\\ 4\\ \cdot\\ \cdot\end{bmatrix}=\begin{bmatrix}\cdot & \cdot & \cdot & & & & & \\ & 1 & 2 & 3 & 0 & 0 & 0 & \\ & 0 & 1 & 2 & 3 & 0 & 0 & \\ & 0 & 0 & 1 & 2 & 3 & 0 & \\ & 0 & 0 & 0 & 1 & 2 & 3 & \\ & & & & & \cdot & \cdot & \cdot\\ & & & & & \cdot & \cdot & \cdot\end{bmatrix}\begin{bmatrix}\cdot\\ 1\\ 2\\ 0\\ 0\\ -1\\ \cdot\end{bmatrix}$$

Rows contain time-reversed copies of impulse response.

Linear system ↔ matrix multiplication

Shift-invariant linear system ↔ 'Toeplitz' matrix

Convolution Theorem 1



Eero Simoncelli, NYU

Convolution Theorem 2

***Multiplication** in the time domain* ⟷ ***Convolution** in the frequency domain*

***Convolution** in the time domain* ⟷ ***Multiplication** in the frequency domain*

But why bother with this seemingly complicated business of transforming?

Convolution Theorem 2

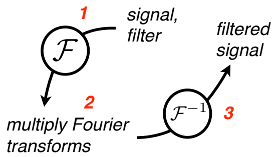
***Multiplication** in the time domain* ⟷ ***Convolution** in the frequency domain*

***Convolution** in the time domain* ⟷ ***Multiplication** in the frequency domain*

But why bother with this seemingly complicated business of transforming?

For large data sets

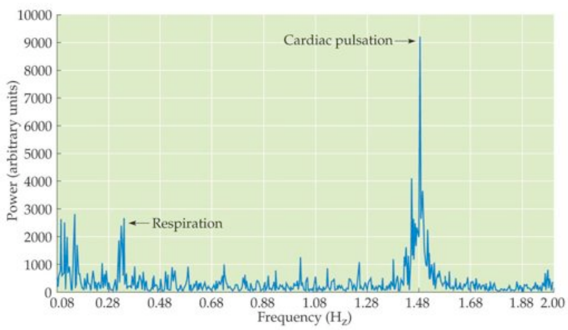
- Convolution is an computationally expensive operation
- FFT / IFFT is very efficient
- Point-by-point multiplication is cheap

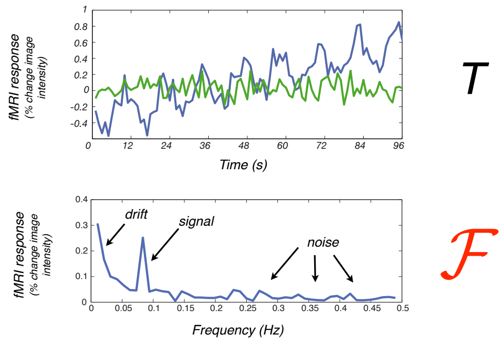


In some cases...

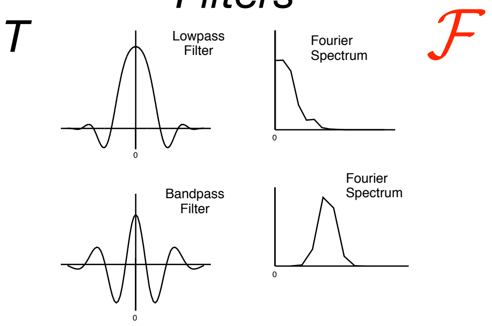
- ... it's easier to see periodic events, e.g. artefacts due to cardiac cycle / breathing in the frequency domain

Lance Armstrong?

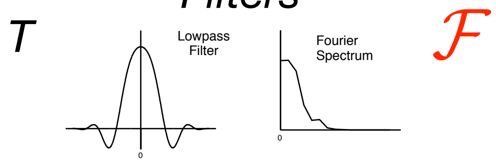




Filters

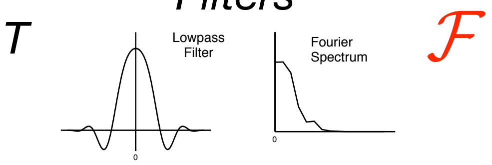


Filters

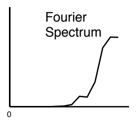


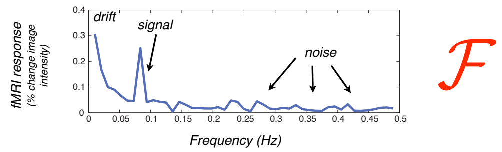
*How would we
make a high-pass filter?*

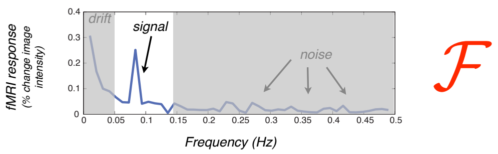
Filters

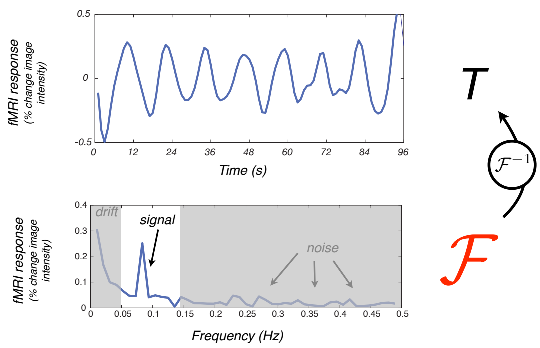


*How would we
make a high-pass filter?*





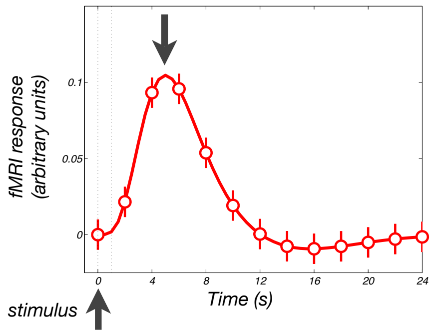




☣ Linear Algebra / FFT

- Eero Simoncelli, NYU
  http://www.cns.nyu.edu/~eero/math-tools/
  contains additional links to www / books
- *MIT OpenCourseWare (video lectures)*
  Mathematics, Gilbert Strang, 18.06 course
- *Linear Algebra and Its Applications*, Gilbert Strang, book

☣ HRF



☣ Glover, 1999

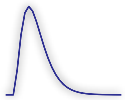
$$H(t) = \left(\frac{t}{d_1}\right)^{a_1} \exp\left(\frac{-(t-d_1)}{b_1}\right) - \left(\frac{t}{d_2}\right)^{a_2} \exp\left(\frac{-(t-d_2)}{b_2}\right)$$

default params [a_1, a_2, b_1, b_2, c] = [6 12 0.9 0.9 0.35]

Glover. Deconvolution of impulse response in event-related BOLD fMRI. *Neuroimage* (1999) vol. 9 (4) pp. 416-29

☣Plot a simple version in Matlab?

$$H(t) = \left(\frac{t}{\tau}\right)^2 \cdot \frac{\exp(-t/\tau)}{2\tau}$$



```
tau = 2; % time constant
delta = 2; % time shift
t = [0:1:30]; % vector of time points
tshift = max(t-delta,0); % shifted, but not < 0
HIRF = (tshift/tau).^2 .* exp(-tshift/tau) ...
/ (2*tau); % function
figure(1), plot(HIRF, 'r'); % plot it
```


Quantifying Signal / Noise

Signal-to-noise ratio (SNR)

raw SNR: used by physicists + engineers to quantify image quality

Intensity in sample (e.g. brain) / *Intensity outside sample (e.g. outside head)*

2000 / ***100***

Contrast-to-noise ratio (CNR)

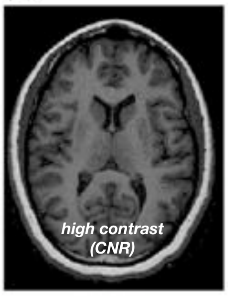
CNR: e.g. how good is T_1 contrast between white matter (WM) and gray matter (GM) – take two small regions of interest

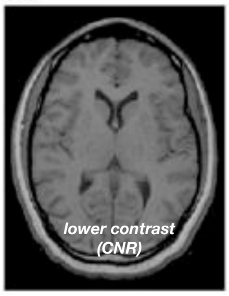
	mean GM	**mean WM**	**noise (σ)**	**cnr**
image 1	*150*	*250*	*100*	*x*
image 2	*60*	*70*	*5*	*?*

Contrast-to-noise ratio (CNR)

CNR: e.g. how good is T_1 contrast between white matter (WM) and gray matter (GM) – take two small regions of interest

	mean GM		**mean WM**	**noise (σ)**	**cnr**
image 1	*150*	*100*	*250*	*100*	*x*
image 2	*60*	*10*	*70*	*5*	*?*



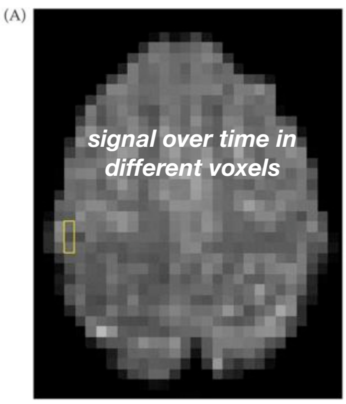


functional signal-to-noise ratio

functional SNR: (sometimes called functional CNR)

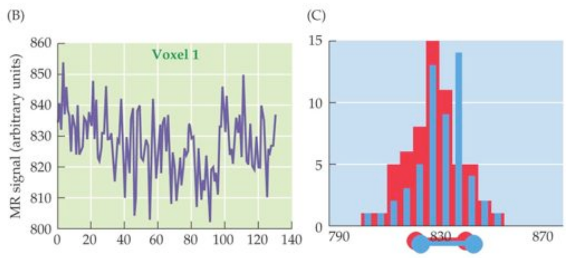
signal: difference between two states of the brain caused by experiment

noise: variability in those states over time...

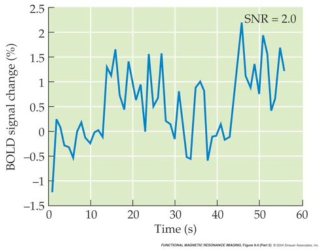


FUNCTIONAL MAGNETIC RESONANCE IMAGING, Figure 9.2 (Part 1) © 2004 Sinauer Associates, Inc.

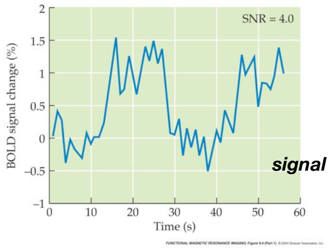
low functional SNR



how to increase functional SNR?

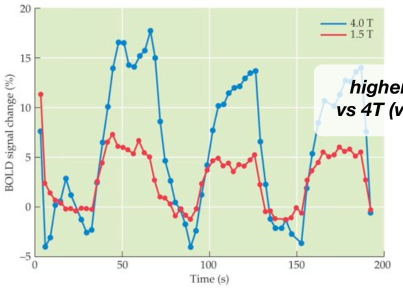


how to increase functional SNR?



signal averaging

how to increase functional SNR?



higher field: 1.5T vs 4T (visual cortex)

Summary

- recap: linear systems
- **Matlab**
- simulated block design data
- drift + (high-frequency) noise
- Fourier domain, convolution
- raw SNR, CNR, functional SNR