Lecture 3: Signal Processing In BCIs

Introduction to Modern Brain-Computer Interface Design

Christian A. Kothe
SCCN, UCSD

Outline

1. The Role of Signal Processing
2. Major Filter Classes
   - Spatial Filters
   - Temporal Filters
   - Spectral Filters
3. A Simple Neurofeedback BCI
4. Prediction Function Notion

Swartz
Center for
Computational
Neuroscience

3.1 The Role of Signal Processing

BCI Theory

- BCI leverages theory from a wide range of fields (Signal Processing, Machine Learning, Statistics, Neuroscience, Control Theory, Information Theory, …)
- A given BCI may be understood from the vantage point of any of these theories
- But no single theory conveniently describes all aspects of a BCI

Signal Processing

- *Digital Signal Processing* is concerned with systems (a.k.a. filters) that transform one signal into another
- *Linear Time-Invariant (LTI) Systems*, including Spectral filters and their optimal design are one of the most developed areas
- *Statistical Signal Processing* and *Adaptive Filtering* are among the advanced areas (Kalman Filter, etc., recursive least-squares)
- *Sparse Signal Processing* (e.g., sparse recovery and compressive sensing) is a new branch with application to BCIs

Signal Processing

- A signal is a mapping from an index set (here discrete time) onto vectors (multichannel samples)
- From the point of view of Signal Processing, a BCI transduces the input signal $x(n)$ (for example EEG) into a control signal $y(n)$
- It is defined by a *transformation rule T*

$$y(n) = T[x(n)]$$

Important System Types

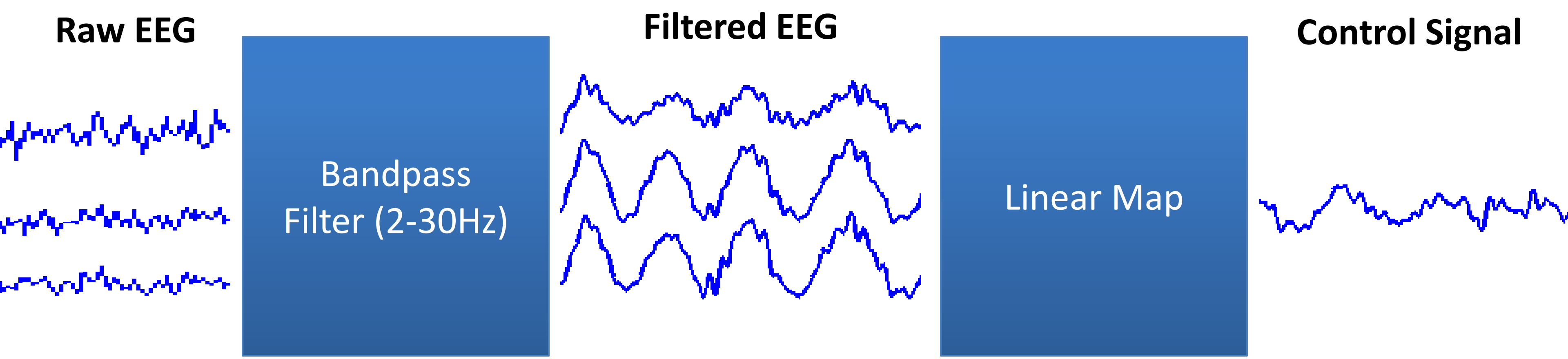
- A system is called ***static*** if the value $y(n)$ at any sample n depends only on $x(n)$, otherwise ***dynamic***.
- A system is called ***causal*** if the output $y(n)$ at any time n only depends on values of $x(m)$ for $m \leq n$, otherwise ***non-causal***.
- A system is called ***time-invariant*** if $y(n) = T[x(n)]$ implies that $y(n-k) = T[x(n-k)]$ for every time shift k, otherwise ***time-variant***.
- A system is called ***linear*** if the equation $T[a_1 x_1(n) + a_2 x_2(n)] = a_1 T[x_1(n)] + a_2 T[x_2(n)]$ holds for all inputs $x_1(n)$ and $x_2(n)$ and all constants a_1 and a_2, otherwise ***nonlinear***.

BCIs Viewed as Filters

- Since BCIs are operated in real time, they are always *causal* systems
- BCIs usually perform temporal filtering, and are therefore *dynamic*
- Some BCIs are *time-invariant*, but adaptive BCIs are not
- Simple BCIs are *linear*, but the vast majority is not
- Since their output is needed at a much lower sampling rate (0.1-60Hz) than the input (250-1000Hz), they are technically *multi-rate* systems

BCI Components as Filters

- BCI components are conveniently described as filters – more so than the entire system itself
- This gives rise to several key categories of filter components

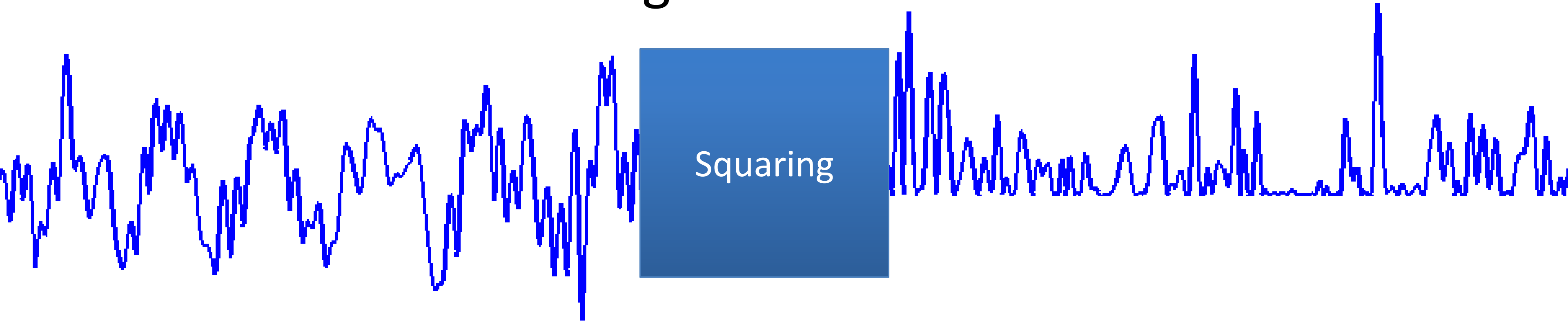


Swartz Center for Computational Neuroscience

3.2 Major Filter Classes

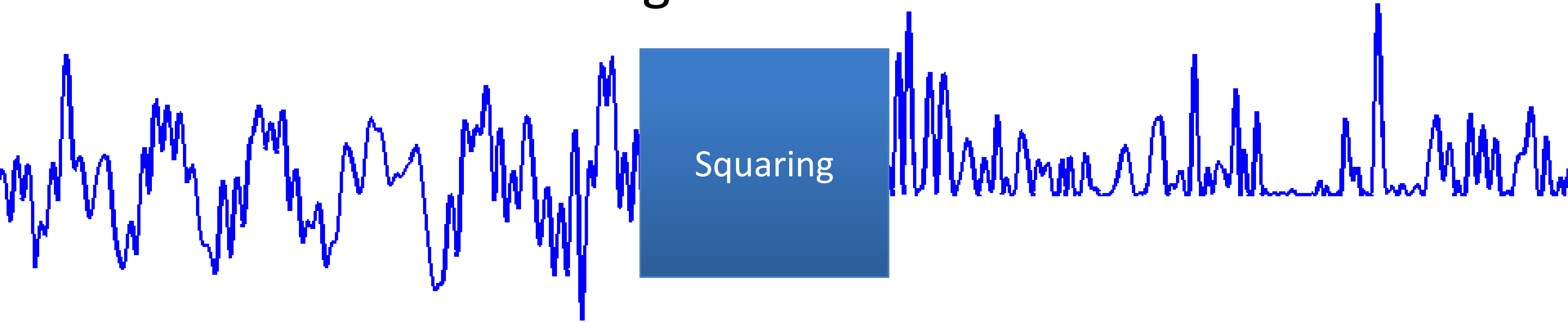
Static Filters

- **Signal Squaring**: $\mathcal{T} \;:=\; y_i(n) \;=\; x_i\,(n)^2$
  - static system, useful step in calculating the variance of the signal



Static Filters

- **Signal Squaring**: $\mathcal{T} \;\coloneqq\; y_i(n) = x_i\,(n)^2$
  - static system, useful step in calculating the variance of the signal



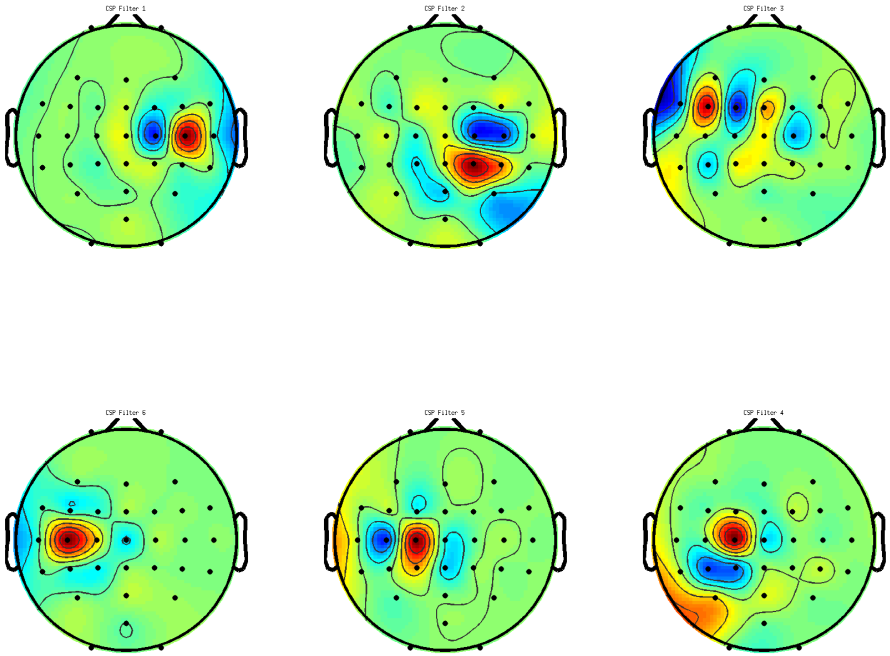
- **Logarithm:** $\mathcal{T} \;\coloneqq\; y_i(n) = \log x_i\,(n)$
  - useful later

Spatial Filters

- Transform a multi-channel signal $X(n)$ such that each $Y(n)$ depends only on $X(n)$; most spatial filters are linear, i.e. $Y(n) = \boldsymbol{M}X(n)$ for some matrix M
- Linear spatial filters can approximately invert volume conduction and remap channel signals to approximate source signals – this is their main use in BCIs
- Examples: Re-referencing, Surface Laplacian, Independent Component Analysis (ICA), Common Spatial Patterns (CSP) – more later

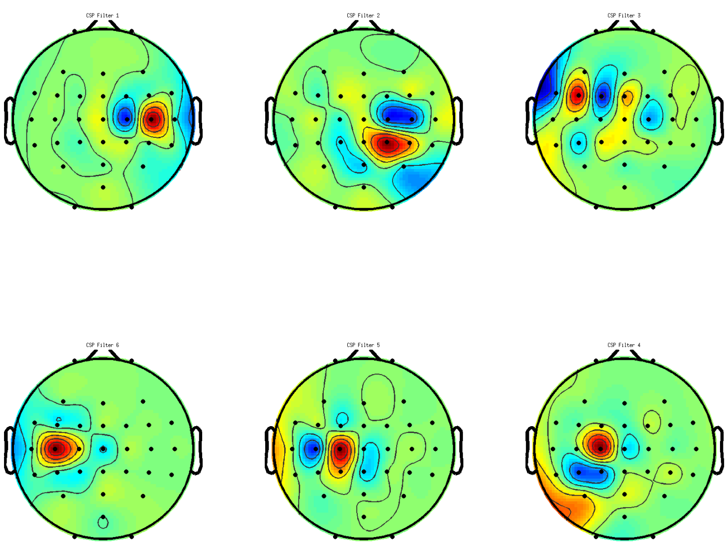
Spatial Filters Visualized

- Spatial filters designed to *recover motor-cortex source activity*, calculated via the Common Spatial Patterns algorithm

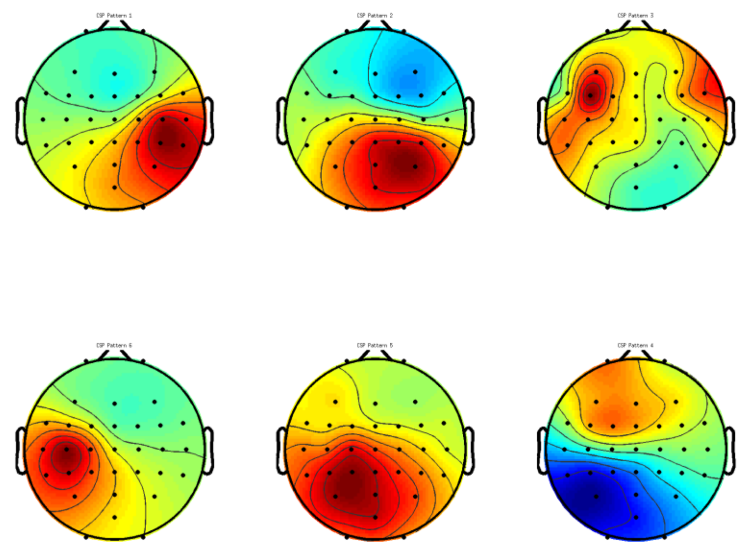


Spatial Filters vs. Forward Projections

- Spatial filters are *not* the same as forward projection maps of some source signal – they are the inverse operation



$S(n) = \boldsymbol{W}X(n)$



$\mathrm{X}(n) = \boldsymbol{W}^{-1}S(n)$

Temporal Filters

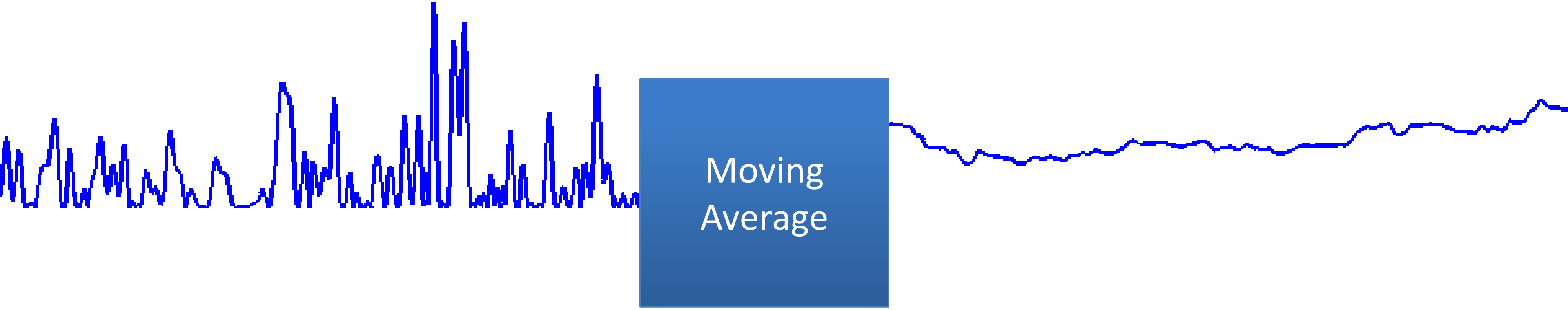
- Transform a multi-channel signal $X(n)$ such that each channel $y_i(n)$ in $Y(n)$ depends only on the channel $x_i(n)$
- They are conceptually orthogonal to spatial filters
- Examples include time windowing, wavelet transform, etc.
- Special case: Spectral filters

Example Temporal Filters

- **Moving Average:**

$$\mathcal{T} \;:=\; y_i(n) \;=\; \frac{1}{m}\sum_{k=0}^{m-1} x_i\,(n-k)$$

  - Effectively a smoothing (low-pass) operator
  - In fact a simple example of a spectral filter



Spectral Filters

- Temporal filters that are designed for their effects on the *spectrum* of the signal
- **Spectrum of a signal**: a representation of the signal as a sum of N sinusoidal components,

$$s(n) = \sum_{k=1}^{N} A_k \sin(\omega_k nT + \phi_k)$$

where A_k is the amplitude of each sinusoid and ϕ_k is its phase.

Spectral Filters

- An equivalent (more common) representation is the Fourier Series representation

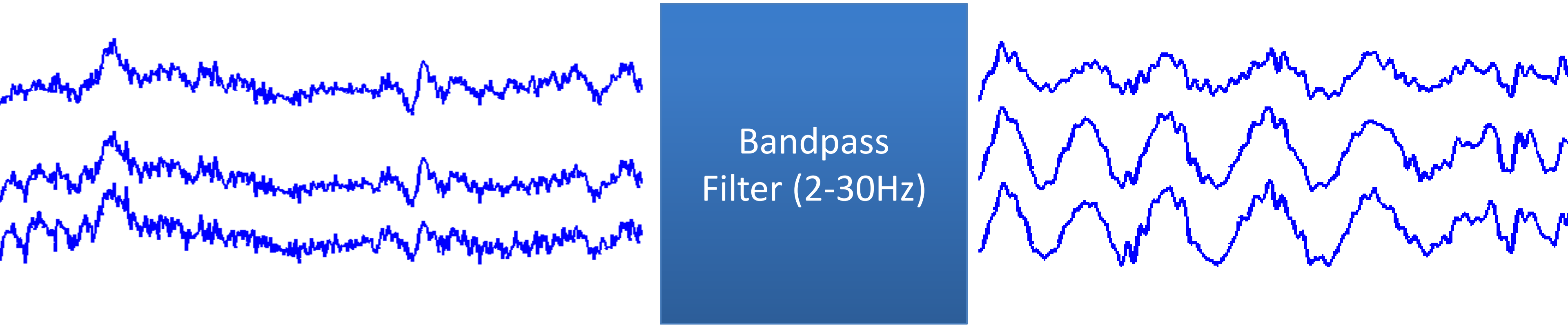
$$s(n) = \sum_{k=0}^{N-1} A_k e^{j2\pi kn/N}$$

  where A_k is now complex-valued and represents both the amplitude and the phase
- This relies on the Euler formula

$$e^{\pm j\phi} = \cos\phi \pm j\sin\phi$$

Spectral Filters

- Examples include: High-pass, Low-pass, Band-pass filters, Notch filters
- Their main utility in BCIs is to isolate oscillations or ERPs of interest



A Key Spectral Filter

- **FIR (Finite Impulse Response) Filter**:

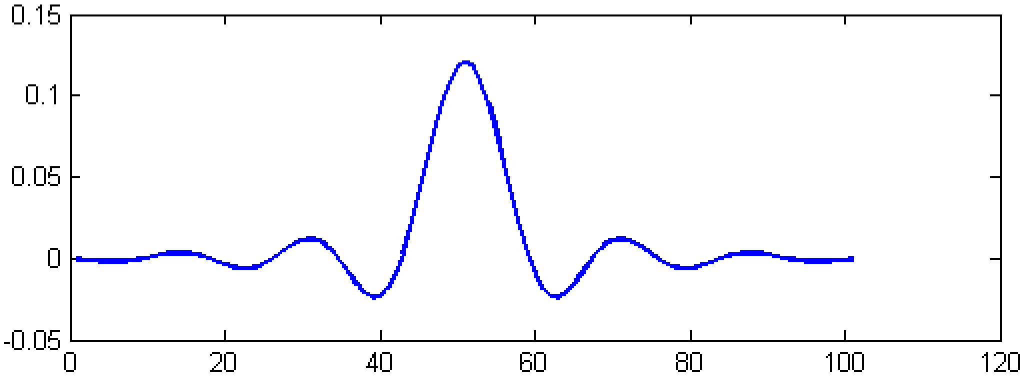
$$\mathcal{T} \; := \; y_i(n) \; = \; \sum_{k=0}^{m} b_k x_i \, (n - t)$$

  - Performs a *convolution* between signal and kernel
  - The trick lies in the coefficients (“kernel”) b_k
  - Can implement any linear time-invariant spectral filter
  - Moving average is a special case

Filter Implementations

- **Lowpass:**

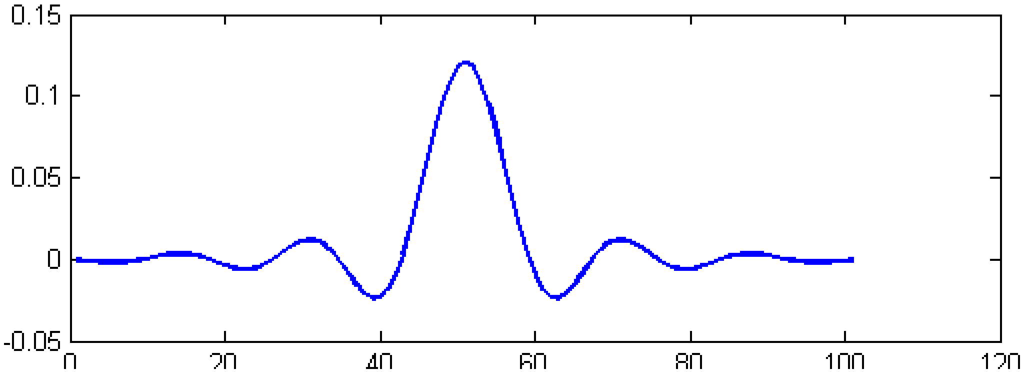
b_k =



Filter Implementations

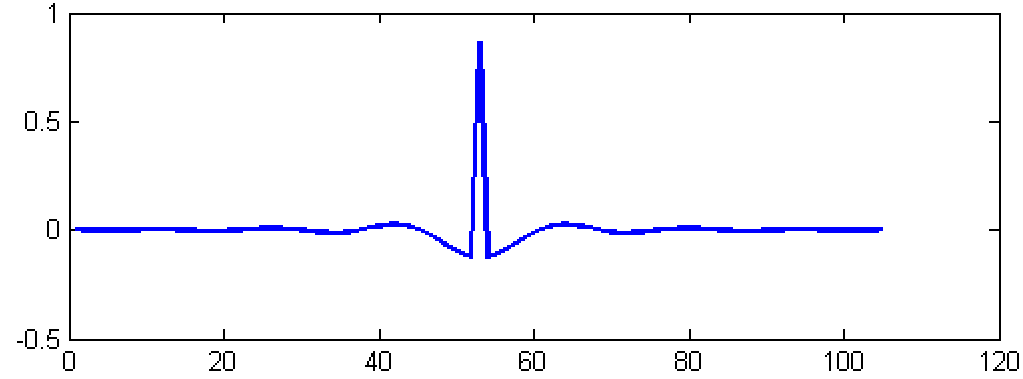
- **Lowpass:**

b_k =



- **Highpass:**

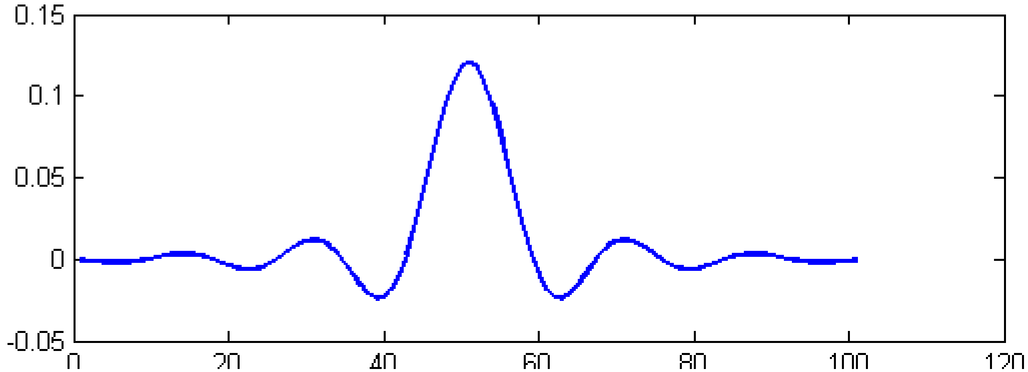
b_k =



Filter Implementations

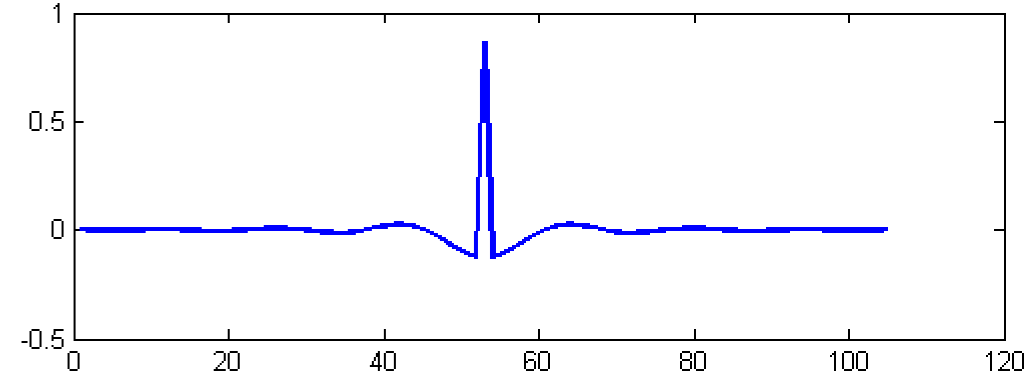
- **Lowpass:**

b_k =



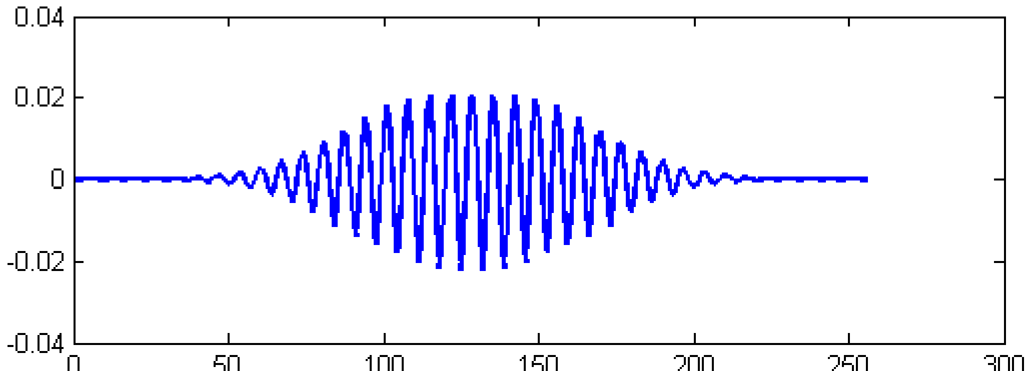
- **Highpass:**

b_k =



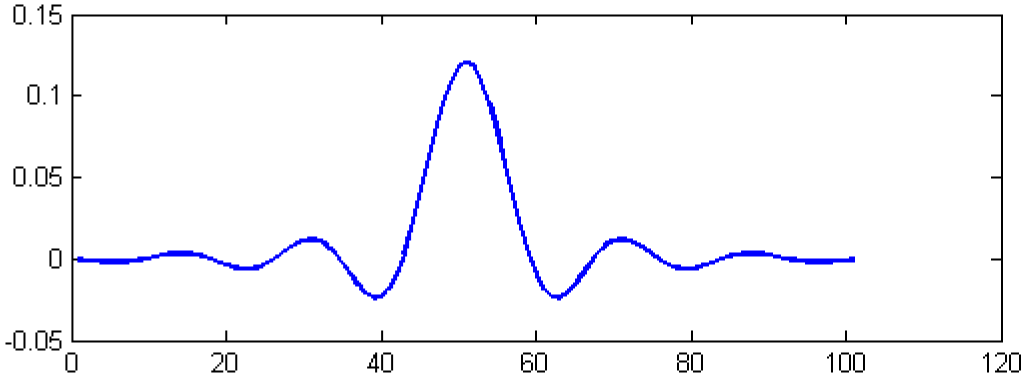
- **Bandpass:**

b_k =

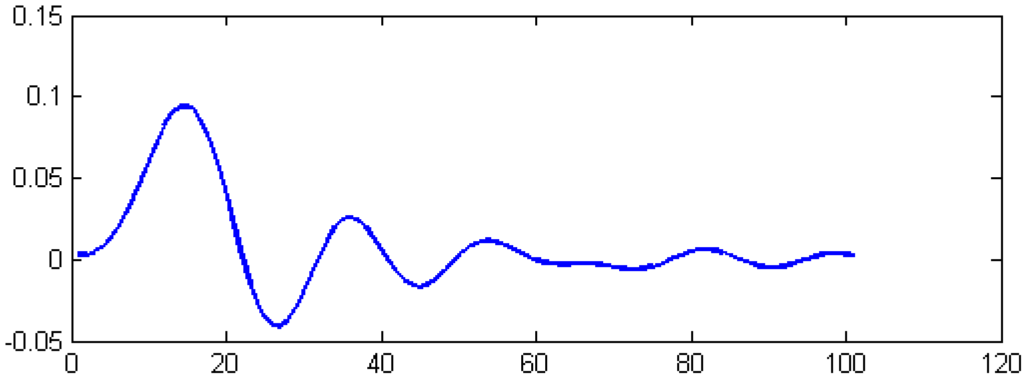


Minimum-Phase Filter Design

- **Linear-Phase Lowpass (uniform lag):**



- **Minimum-Phase Lowpass (minimal lag):**



FIR Filter Design in MATLAB

- Various criteria for filter kernel design (given a desired frequency response):
  - Least-Squares error: `firls`
  - Minimax error (Parks-McClellan): `firpm`
  - Using the Fourier transform: `fir2`
- Choice of a reasonable filter order (length):
  - `firpmord`
- Minimum-phase filter design (using Cepstral analysis): `rceps`

Other Filters

- **Spatio-Temporal Filters** are also used, but have too many degrees of freedom to be hand-designed; usually the result of an adaptive procedure
- **Spectral Transforms**, which transform between the time and spectral representation of a signal are used frequently as intermediate stages
- **Rate-changing Filters** such as resampling are useful to manage computational costs

Swartz
Center for
Computational
Neuroscience

3.3 A Simple Neurofeedback BCI

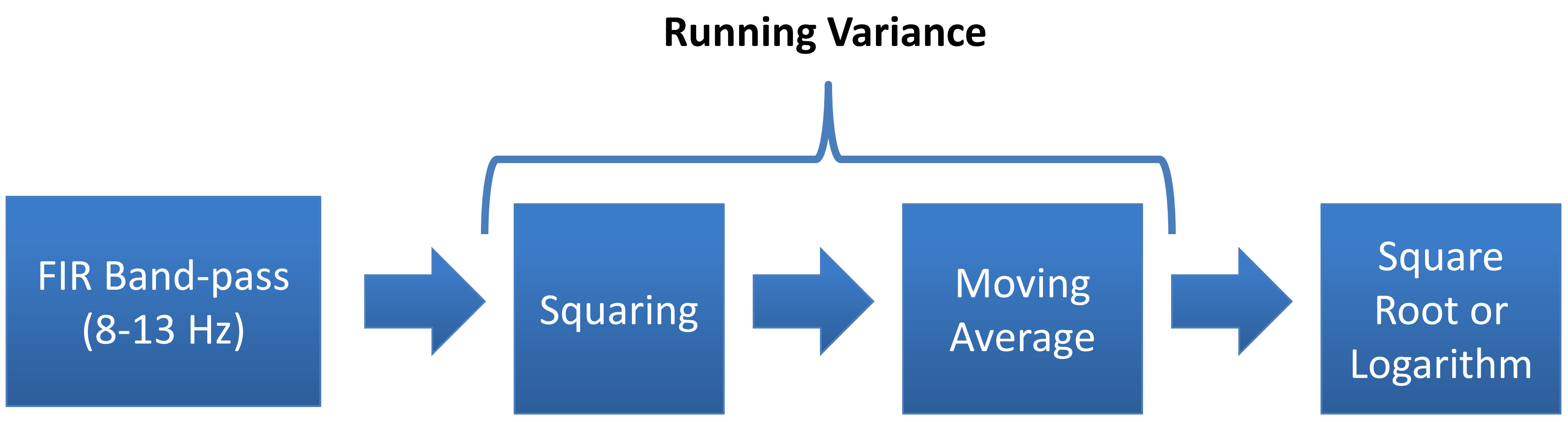
A Simple Neurofeedback BCI

- Feed back the amplitude of a brain idle oscillation (e.g. 10 Hz alpha for relaxation) to the user/subject
- Also other processes conceivable

source: psychful.com

A Simple Neurofeedback BCI

- Feed back the amplitude of a brain idle oscillation (e.g. 10 Hz alpha for relaxation) to the user/subject
- Can be implemented using discussed tools:

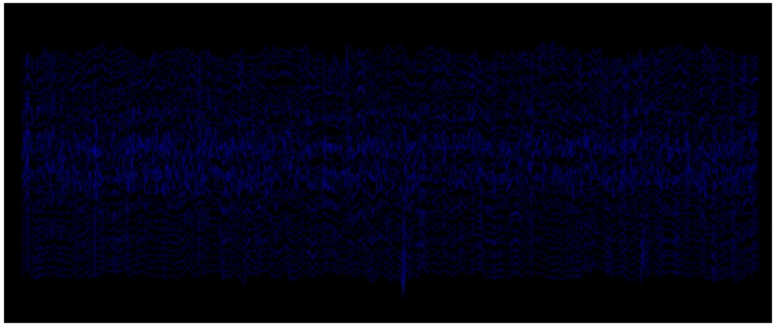


3.4 Prediction Function Notion

Alternative to the Signal Processing Framework

- A BCI with a limited memory of the past could also be viewed as a mathematical mapping *f*:

y = f(**X**); **X**=

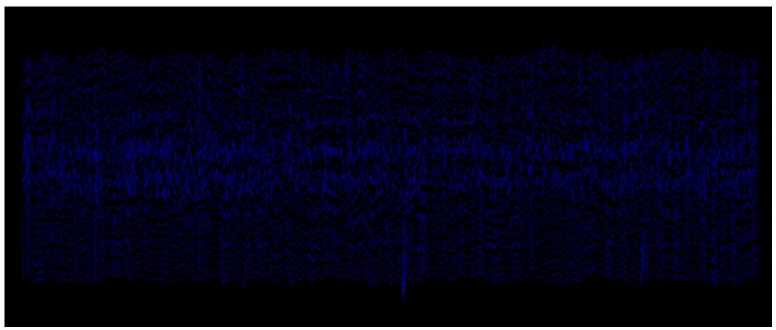


y= “subj. excited” (+1)
“subj. not excited” (-1)

Alternative to the Signal Processing Framework

- A BCI with a limited memory of the past could also be viewed as a mathematical mapping *f*:

y = f(**X**); **X**=



y= "subj. excited" (+1)
"subj. not excited" (-1)

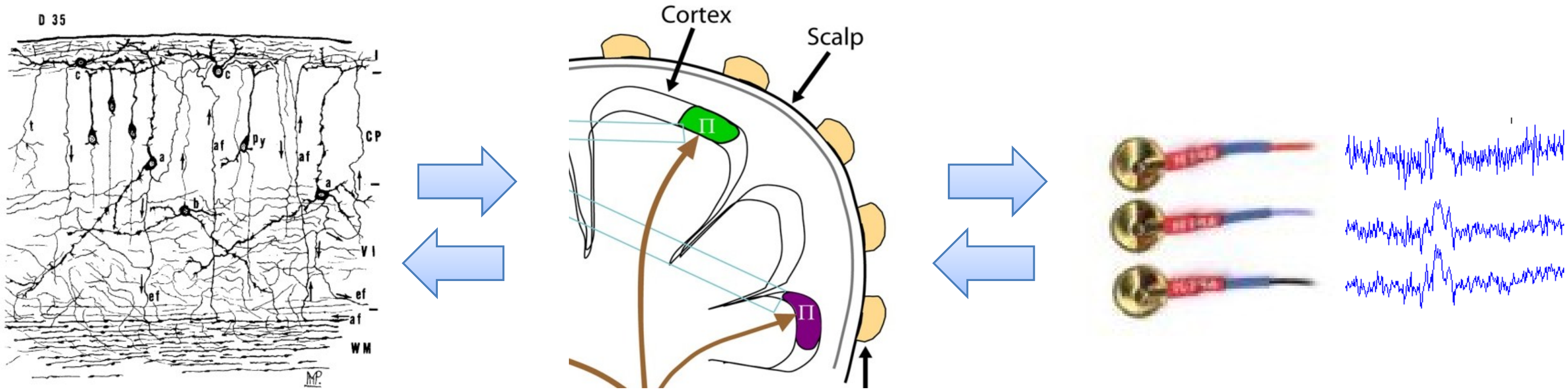
- The functional form is arbitrary, for example

$$y = \mathrm{sign}(\mathrm{var}(\boldsymbol{W}\boldsymbol{X}) + b)$$

- The mapping involves unknown parameters, here **W** and b

Functional Form

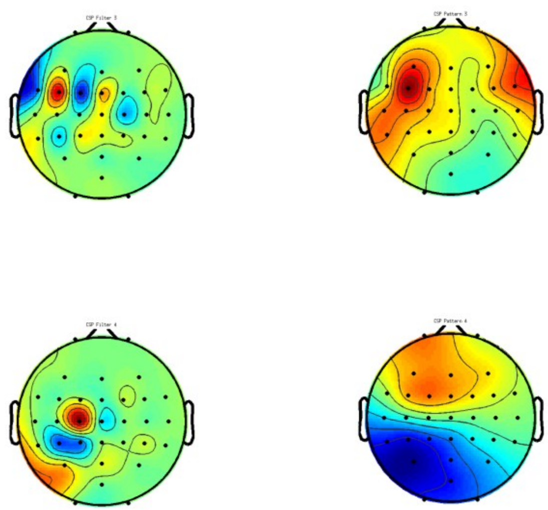
- Reflects the relationship between observation (data segment X) and desired output (cognitive state parameter y)
- Based on some assumed generative mechanism (forward model) – or ad hoc



- Note: Functional form is the inverse mapping!

Core Ingredient: Spatial Filter

- Linear inverse of volume conduction effect
  $\boldsymbol{X} = \boldsymbol{AS}$ (forward)
  $\boldsymbol{S} = \boldsymbol{WX}$ (inverse)



W **A**

Full Examples (Details Later)

- Inverse mapping from filtered source time courses to latent cognitive state, e.g.:

$$y = \boldsymbol{\theta}\, \mathbf{vec}(\boldsymbol{W}\boldsymbol{X}) + b$$ **(linear)**

$$y = \boldsymbol{\theta}\, \mathbf{vec}(|(\boldsymbol{W}\boldsymbol{X})\boldsymbol{T}|) + b$$ **(nonlinear…)**

Neurofeedback BCI in Functional Style

- Performed as a mapping of a sliding window **X** onto the output y:

$$f(\boldsymbol{X}) \quad := \quad y = \log var(\boldsymbol{XT})$$

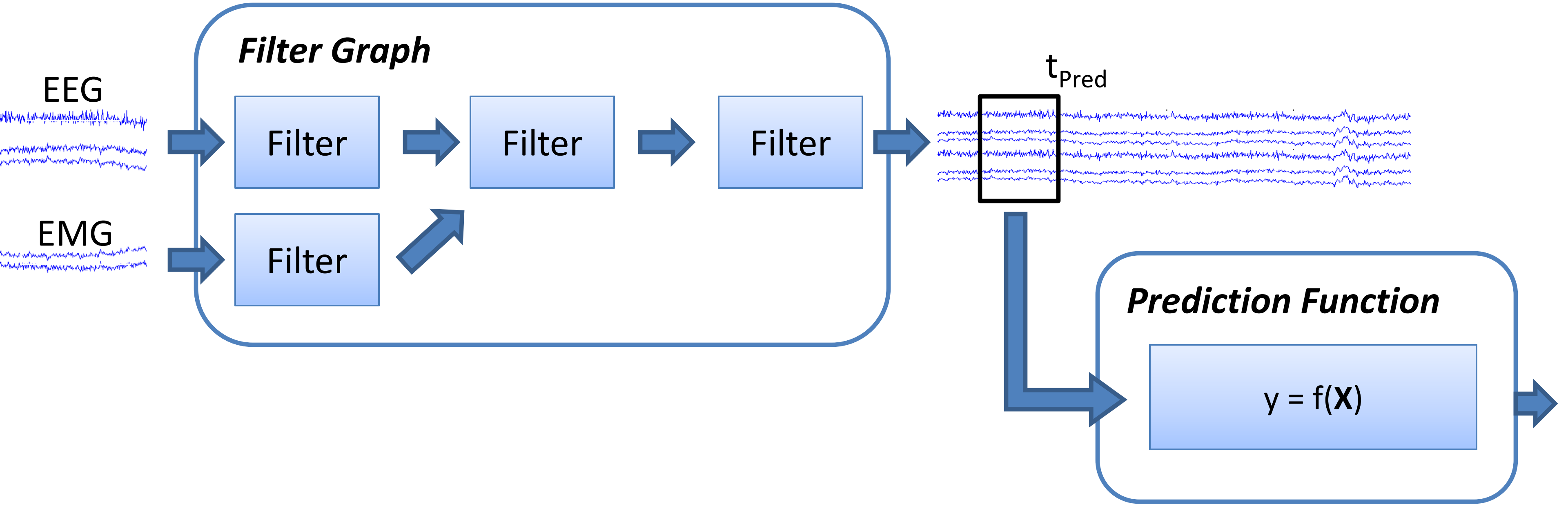
- $\boldsymbol{T}$ implements a temporal filter, written as a matrix multiplication (each column = shifted filter kernel)

In Comparison…

- **Main drawback** of the pure mathematical form compared to the signal processing approach:
  - The entire input window **X** is *re-processed* for each desired output value y
  - Especially bad if the window **X** moves only by a few samples between evaluations of f
  - In contrast, most signal processing methods are *incremental* or *recursive* (e.g. FIR or IIR)
- **Main benefit** is the relative conceptual simplicity

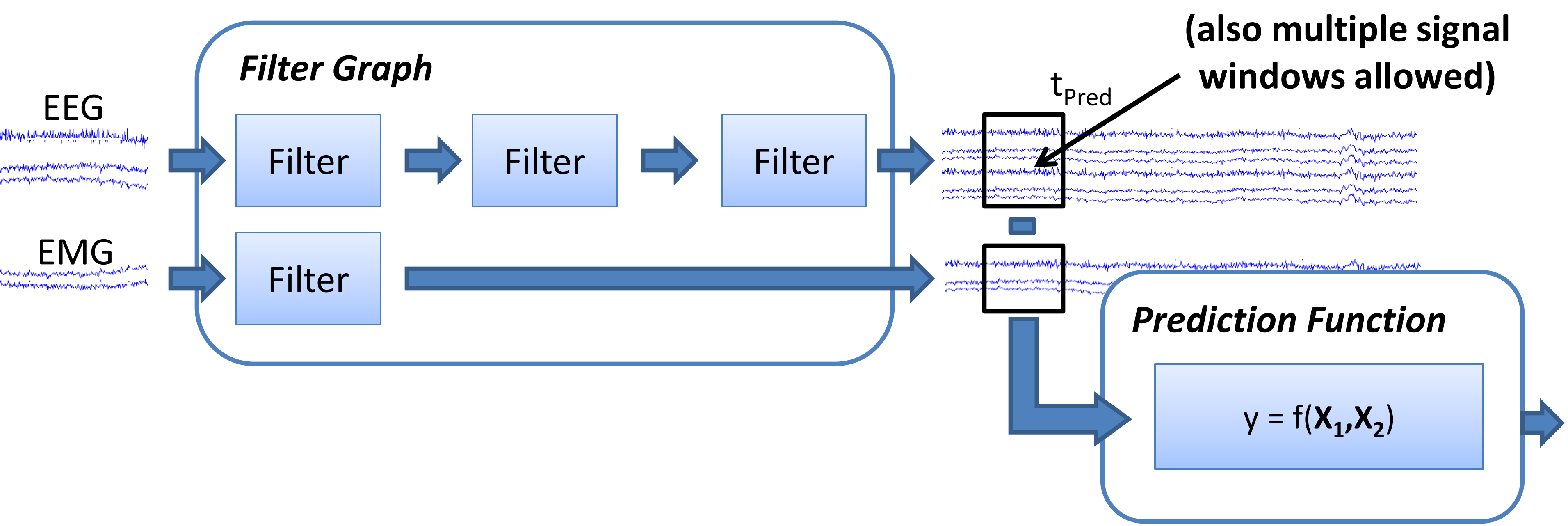
In Combination

- Both frameworks are complementary, rather than contradictory, and are in practice often used *in combination*
- Prediction function is queried on demand



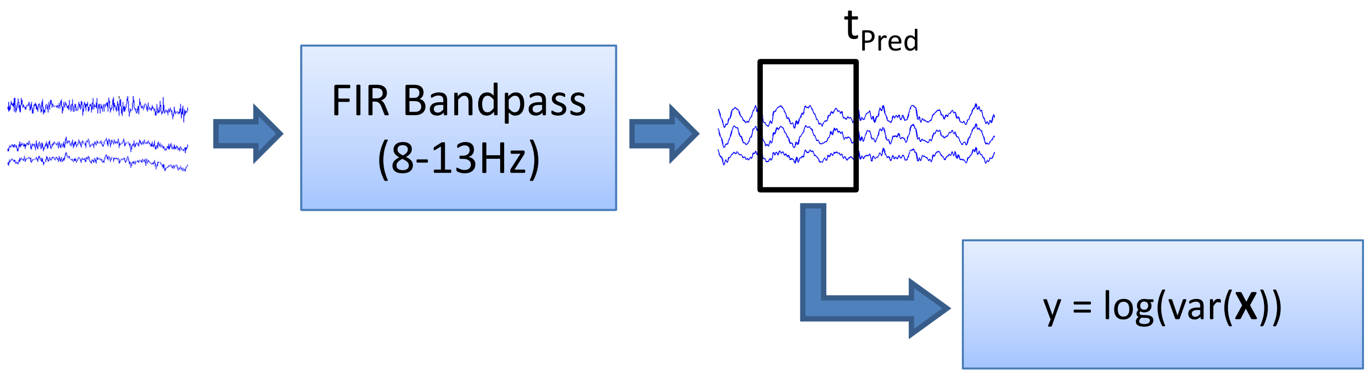
In Combination

- Both frameworks are complementary, rather than contradictory, and are in practice often used *in combination*
- Prediction function is queried on demand



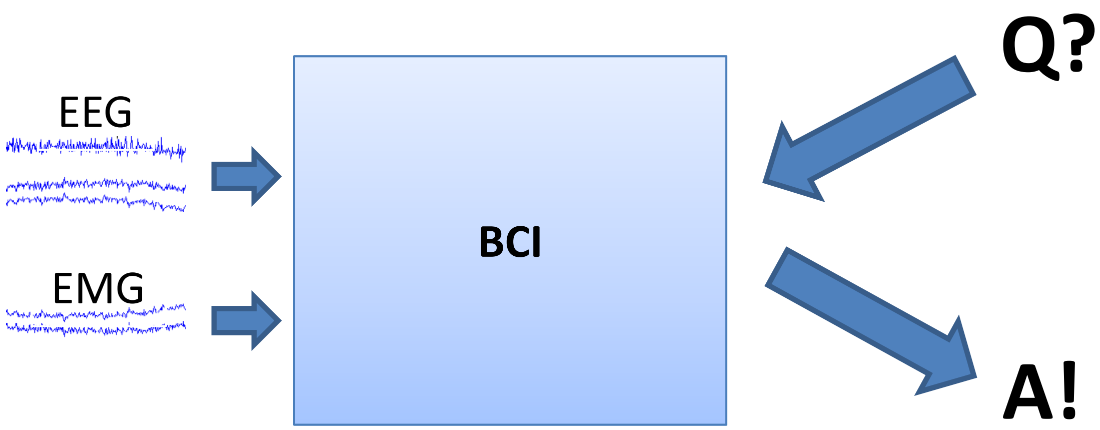
Neurofeedback Using the Combined Approach

- Computationally costly spectral filtering is done in the signal processing portion
- Lightweight predictive mapping is done at lower rate in the functional portion



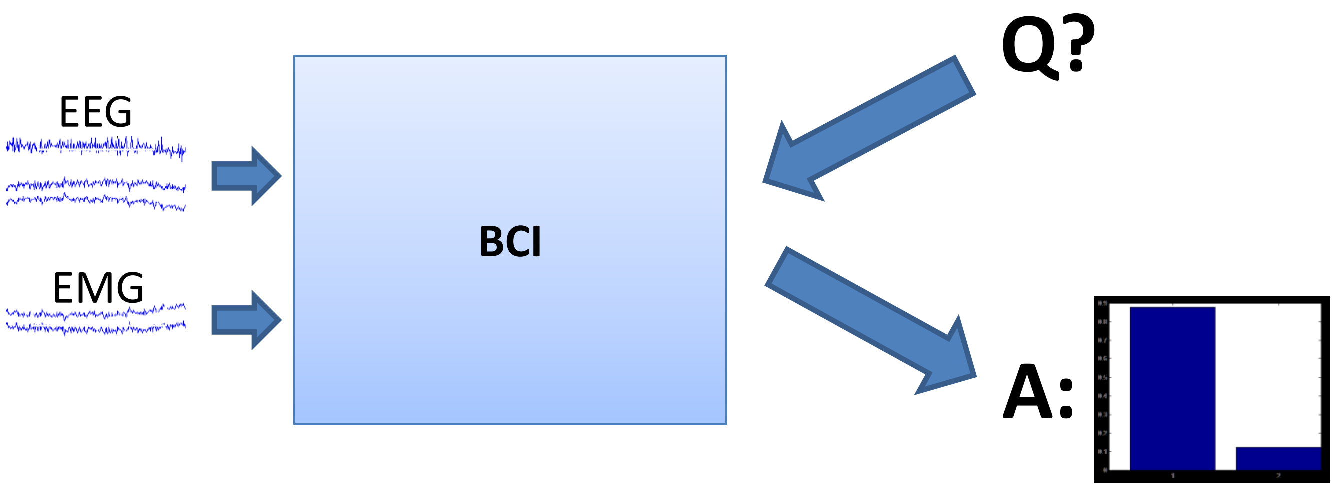
Outside View

- Implemented this way, BCIs act as an *oracle* that consumes one or more multi-channel signals and can respond to queries about a pre-defined question



Outside View

- Implemented this way, BCIs act as an *oracle* that consumes one or more multi-channel signals and can respond to queries about a pre-defined question
- Note: in modern BCIs the output is often a discrete probability distribution



Swartz
Center for
Computational
Neuroscience

L3 Questions?