Introduction to BCI: Decoding(1) - Artifacts and filtering

J.Farquhar

Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour

September 13, 2011

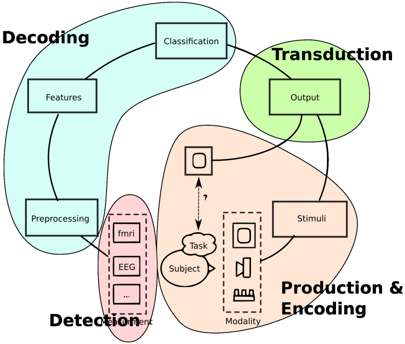
Outline

Outline

How does a BCI work?

Fundamentally, BCI is a simple(?) engineering problem;

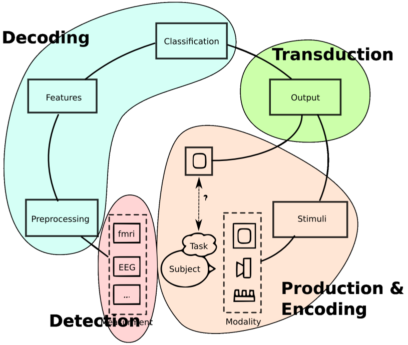
1. Signal Production: Get the person to produce a strong brain signal, either by performing an explicit mental-task, or through normal mental processes
2. Detection: Build a machine able to measure the properties of their brain, e.g. EEG, MEG, fMRI
3. Decoding: Build a machine able to decode the measurements to deduce the users mental state
4. Transduction: Communicate the mental-state to the outside world



How does a BCI work?

Fundamentally, BCI is a simple(?) engineering problem;

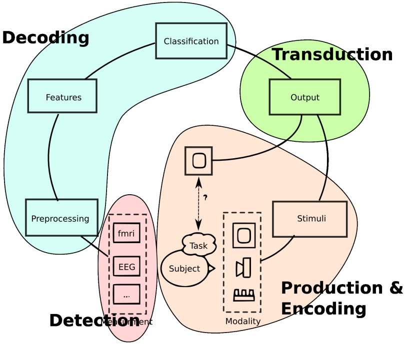
1. Signal Production: Get the person to produce a strong brain signal, either by performing an explicit mental-task, or through normal mental processes
2. Detection: Build a machine able to measure the properties of their brain, e.g. EEG, MEG, fMRI
3. Decoding: Build a machine able to decode the measurements to deduce the users mental state
4. Transduction: Communicate the mental-state to the outside world



How does a BCI work?

Fundamentally, BCI is a simple(?) engineering problem;

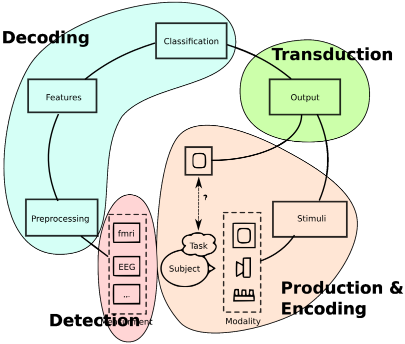
1. Signal Production: Get the person to produce a strong brain signal, either by performing an explicit mental-task, or through normal mental processes
2. Detection: Build a machine able to measure the properties of their brain, e.g. EEG, MEG, fMRI
3. Decoding: Build a machine able to decode the measurements to deduce the users mental state
4. Transduction: Communicate the mental-state to the outside world



How does a BCI work?

Fundamentally, BCI is a simple(?) engineering problem;

1. Signal Production: Get the person to produce a strong brain signal, either by performing an explicit mental-task, or through normal mental processes
2. Detection: Build a machine able to measure the properties of their brain, e.g. EEG, MEG, fMRI
3. Decoding: Build a machine able to decode the measurements to deduce the users mental state
4. Transduction: Communicate the mental-state to the outside world



The need for signal-processing

Just to re-iterate

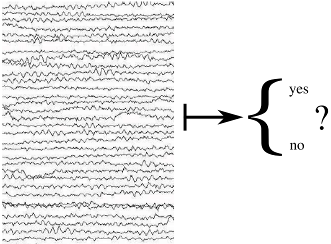
Defn. BCI

a system which allows someone to communicate information about their mental state without the use of the peripheral nervous system.

Note the emphasis. That means we can't use:

- muscles
- peripheral nerves, e.g. motor neurons
- machine artifacts

Decoding the neural code



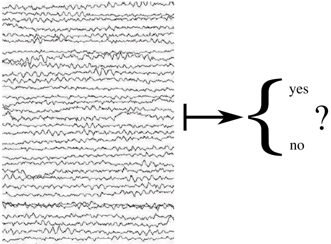
Unfortunately

- In general, we don't know how the users intentions are encoded in the neural signal
- we don't know how the signal may be encoded for this individual

Further:

- non-brain sources generate much stronger signals than the brain
- Thus we must remove them to be sure our BCI has a B in it..
- and isn't a: MMI, EMI, CCI etc..

Decoding the neural code



Unfortunately

- In general, we don't know how the users intentions are encoded in the neural signal
- we don't know how the signal may be encoded for this individual

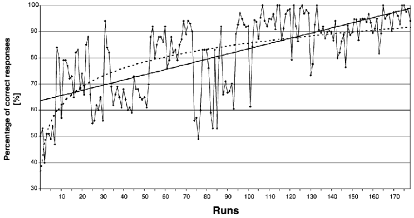
Further:

- non-brain sources generate much stronger signals than the brain
- Thus we must remove them to be sure our BCI has a B in it..
- and isn't a: MMI, EMI, CCI etc..

Decoding Options

Thus we have two options:

1. train the user to generate a strong known signal, e.g. SCP, μERDS
2. train the machine to identify and extract the signature of this users mental state..



Thus:

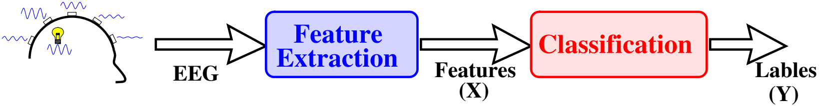
We use signal-processing and machine learning to automatically decode the neural code for each new subject.

Learning Objectives

- Understand why pre-processing is necessary in BCI
- Describe the main sources of artifacts and their effects on the recorded signal; i.e. muscles, eyes, movement, hardware, line-noise etc.
- Describe the main types of pre-processing and what they are useful for; re-referencing, spectral-filtering, artifact-detection/rejection/removal
- Understand what is shown in the temporal, spectral and time-frequency representations of a signal
- Understand how signal filtering in these different representations can be used to suppress the noise in measured data, and hence make a signal more visible
- Understand how knowledge of the signal characteristics allows one to design signal filters to improve signal-to-noise

Outline

Decoding Steps



The decoding process consists of 2 main steps:

1. Pre-processing and feature extraction - remove noise and focus on signal-characteristics of interest
2. Classification - learn a mapping from pre-processed features to decoded-mental states

Typical pre-processing pipeline

1. Pre-processing
   - re-reference
   - detect+reject bad-channels/epochs
   - data-cleaning - remove known artifact sources
   - spectrally/temporally filter
2. Feature Extraction : extract problem specific features based on knowledge of the signal characteristics, e.g. PSD for μ-ERD/S
3. Classification : train a classifier with the extracted features to learn the subject-specific signal

Golden rule of pre-processing

<rant on>

Don't leak label information into your data.

- BCI signal is very small
- Noise is big $\rightarrow$ after removal can still be larger than BCI signal
- class aware pre-processing can leak class information
- clever classifier will use this leaked information if available

The Golden Rule of Pre-Processing

Use exactly the same pre-processing for all epochs.

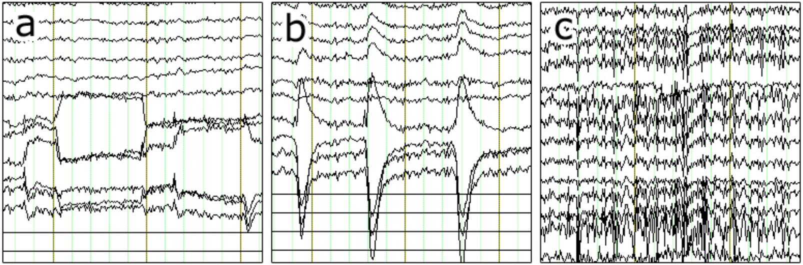
i.e.

- epoch-by-epoch pre-processing
- all-epochs pre-processing

</rant off>

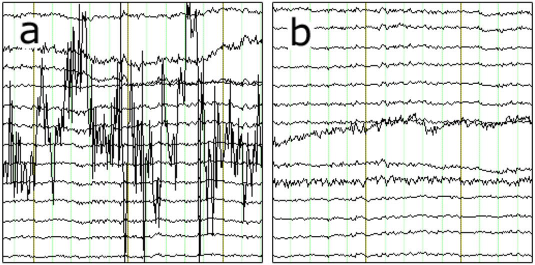
Outline

Example Artifacts (1)



- a Eye movement
- b Eye blink
- c Muscle tension

Example Artifacts (2)

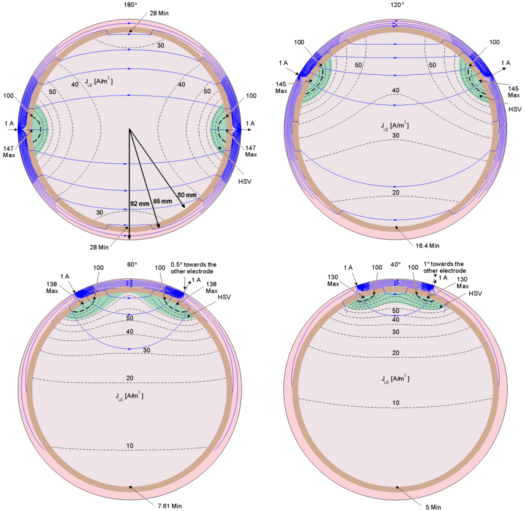


a bad channel
b baseline drift

Outline

Reference Sensitivity

- EEG measures voltage differences between electrode and reference
- Thus, EEG signal is sensitive to selection of reference location



Re-referencing

- Amplifier internally uses a particular reference/ground
- re-reference in s/w to change sensitivity region of each electrode
- assume the recorded signal at each electrode the combination of true signal, s, and common artifact, n, i.e.

$$\mathbf{x}(t) = \mathrm{s}(t) + n(t)$$

- thus, re-referencing to a new reference channel, $r(t)$, can remove the artifact

$$\mathbf{x}_r(t) = \mathrm{s}(t) + n(t) - r(t)$$

- re-referencing changes how your data looks
- removes common parts of the reference, e.g. $n(t)$, from $\mathbf{x}$,
- spreads non-common parts of the reference into all electrodes

Thus

ideal reference contains only the common noise component

Commonly used references (1)

- Ear - far from brain
- (Linked)-mastoids - far from brain
- Common-average / Common-mode
  - assuming signals s are zero-mean,

$$< \mathbf{x}(t) >=< \mathbf{s}(t) > - < n(t) >= n(t)$$

  - Note: non-zero-mean part of signal gets spread over all electrodes

Commonly used references (2)

- local-reference
  - assuming signals s_l is locally zero-mean, but noise is not

$$\mathbf{x}_l(t) = \mathbf{s}_l(t) - n_l(t)$$

  - where l is some local group of electrodes
  - thus, $< \mathbf{x}_l(t) > = < \mathbf{s}_l(t) > - < n_l(t) > = n_l(t)$
  - N.B. vol-conducted signal nearby sources also tends to be locally non-zero mean, thus improves spatial-selectivity
- surface-lapacian (SLAP)
  - special form of local-reference
  - estimate of the local-divergence of the scalp current
  - shown to be good approx. to dura potential [Nunez 1987]
  - significant increase in spatial selectivity
  - estimated using spline interpolation, see [Perrin 1989]

Outline

Data-Cleaning

- Detection – find where the artifacts are in the data, easy->hard
- Rejection – delete data containing artifacts, easy
- Removal – remove the artifact leaving signal intact, hard

Mk 1 - eyeball

Visually scan the data and hand mark artifacts

- Slow
- Tedious and prone to errors
- good as a final check

Outlier Detection

- Assume most data is artifact free
- Look for data which is significantly different from the rest
- treat these as artifacts

Note

need some measure of similarity between data-segments to assess significance

Example: variance based bad-electrode detection

- bad channels tend to have more noise than good ones
- this noise increases the variance/power of the data, i.e. variance is our similarity measure

Processing Steps:

1. estimate the variance of each electrode
2. compute the mean and variance of these estimates
3. electrodes more than (say) 4-std-deviations from mean are detected as bad

Note

Note: Fieldtrip has a mode for performing this type of analysis visually jointly over electrodes and epochs

Template Matching Detection

- Some artifacts have a particular shape
  e.g. eye-blink, saccads
- look for that shape in data to detect the artifact

Note

Need a good (subject-specific) template shape

Note

Can match templates in frequency-domain, or over multiple electrodes

Template Matching

Processing Steps:

1. visually select a set of example artifacts
2. average these together to generate a template artifact, $a(t)$
3. convolve the template with the data to compute a similarity measure

$$\mathrm{sim}(t) = \sum_{\tau} x(t+\tau)a(\tau)$$

4. peak-detect & threshold to detect artifacts

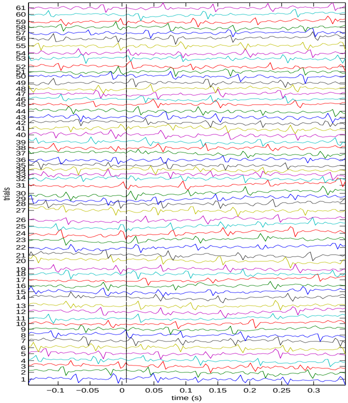
animation

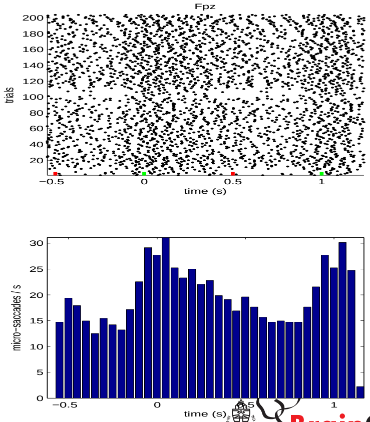
Note

Related idea is to train a classifier to recognise the artifact

Saccade detection

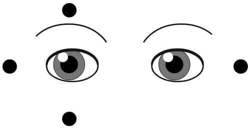
- saccades have a characteristic shape+location
- anywhere this shape occurs at that location is an artifact





Sensor Based Detection

- Many artifacts are generated by known physical processes,
  e.g. eye-blinks, arm-movements, external-stimulation
- record the generating process to identify the artifact



Example: eye-blink detection

- add extra EOG channels to the montage
- threshold (or template match) the EOG data to detect eye-blinks

Note

Record everything you can, e.g. eye, muscle, heart, audio/video stimulus, so you identify causes of artifacts

Artifact Removal

Now that we have detected our artifacts we must remove them from the data

- Rejection – just throw away the data with artifacts
- Removal – attempt to clean the artifacts from the data

Linear Artifact removal

- say we have detected an artifact and extracted a template, n
- this could be from any of the methods presented above, i.e. template match, extra-sensor, virtual-sensor etc.
- assume the recorded signal is a weighted combination of true signal, s, and the artifact, n, i.e.

$$x(t) = s(t) + an(t)$$

- then by simply subtracting $an(t)$ from x we remove the artifact!
- the problem is we don't know the weighting a
- the weight of a source over all electrodes is called its spatial pattern

Two main ways to estimate a

1. correlation-based
2. volume-conduction based

Correlation-based artifact removal

- assume the signal, s, is un-correlated with the artifact, n, i.e. $s^\top n = 0$, then
$$x^\top n = (s + an)^\top n = an^\top n$$
- Hence,
$$a = \frac{x^\top n}{n^\top n}$$
- Combining with artifact removal gives,
$$s = x - an = x - n\frac{n^\top x}{n^\top n} = \left(I - \frac{nn^\top}{n^\top n}\right) x$$

Note: This operation is also called deflation, and de-correlation of the 2 signals x and n.
Example: EOG electrode based eye-blink removal

- use the raw EOG data as the noise signal n to subtract

Volume conduction-based artifact removal

- Vol-conduction means that the weight *a* depends (mainly) on the distance between the artifact source and the electrode
- Thus *a* can be computed given an assumed artifact source location using a forward model

Example: Common-Average-Referencing

- re-referencing to common average is a special case
- assume artifact far enough away that all electrodes have the same weight *a*
- then because all the other sources will tend to cancel out
- averaging the electrode activations leaves only the weighted artifact
- thus, subtracting average activation removes this artifact

Note

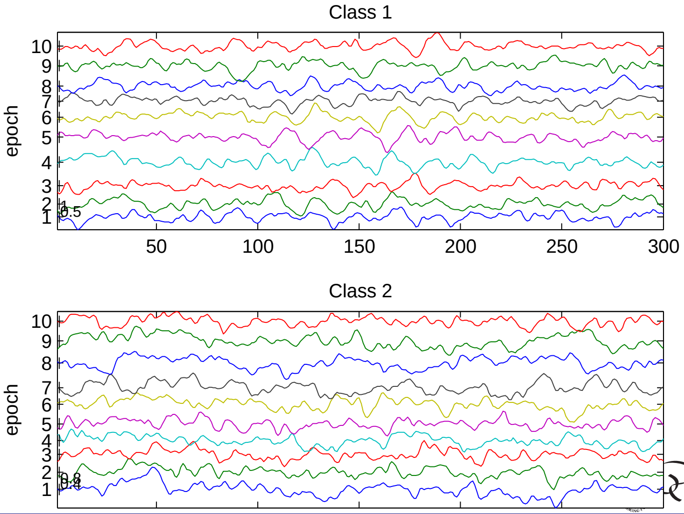
The spatial patterns computed by some Spatial-filtering methods, e.g. PCA, ICA, CSP, directly gives us the electrode weights.

Outline

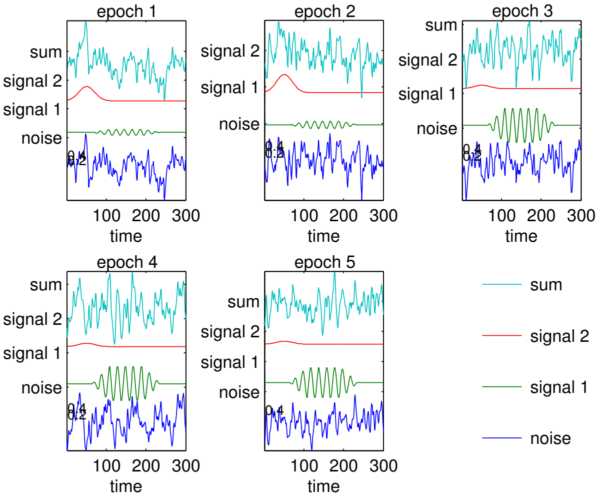
Introduction to BCI signal pre-processing

- BCI data = signal + noise
- usually the signal is much smaller than the noise
- signal processing techniques are used to suppress the noise to make the signal detectable
- most common technique is to use a spectral decomposition of the signal

Example Signals: Simulated 1-ch BCI



Simulated 1-ch BCI Signals



Spectral Decomposition of Signals

- Re-represents the data as a sum of sine and cosine terms

$$x(t) = \sum_{\nu=0:1/T:F/2} f_{\cos}(\nu)\cos(2\pi\nu t) + f_{\sin}(\nu)\sin(2\pi\nu t)$$

- or equivalently, as sum of phase-shifted sines

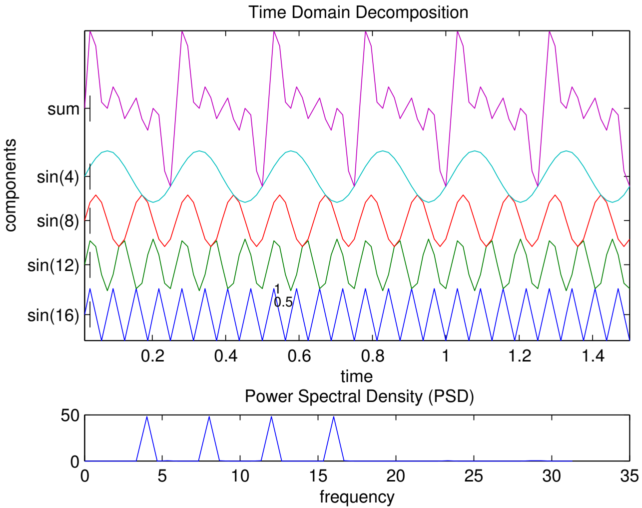
$$x(t) = \sum_{\nu=0:1/T:F/2} f_A(\nu)\sin(2\pi\nu t + f_\omega(\nu))$$

- more commonly written using complex numbers as:

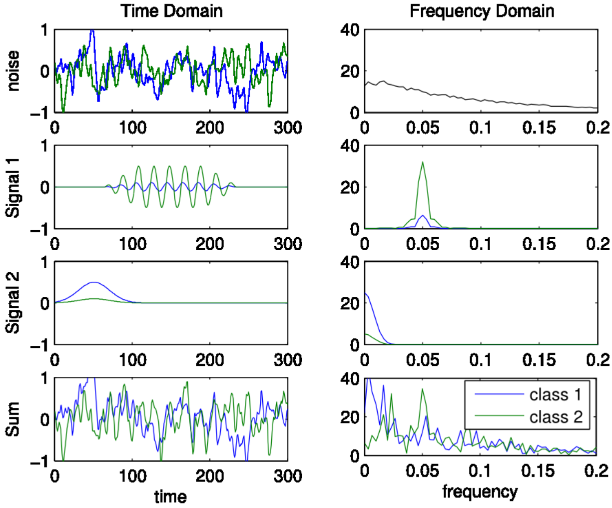
$$x(t) = \sum_{\nu=-F/2:(1/T):F/2} f(\nu)\exp(i2\pi\nu t)$$

where $f(\nu) = f_A(\nu)\exp(\mathrm{sgn}(\nu) i f_\omega(\nu))$
- Frequency Resolution: determined by signal duration, $\Delta f = 1/T$
- Frequency Range: determined by sampling rate, $0 : F/2$

Spectral decomposition of signals - easy



Spectral decomposition of signals - sim data



Time-Frequency Decompositions

Time-Frequency Decomposition

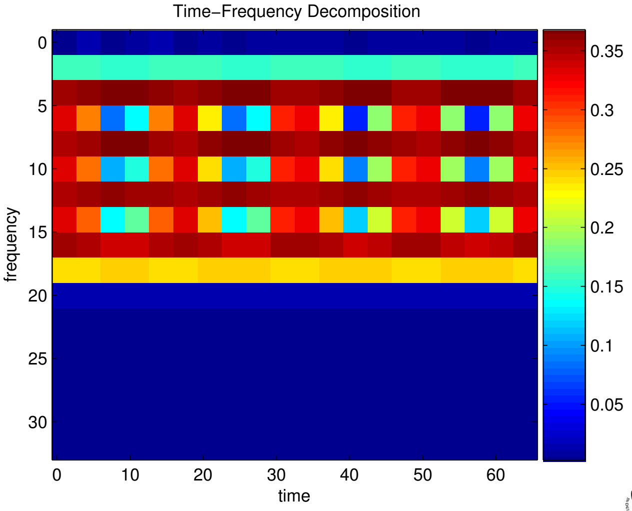
Given information about the magnitude (and possibly phase) of the signal at a particular frequency and time.

Many possible ways of computing this:

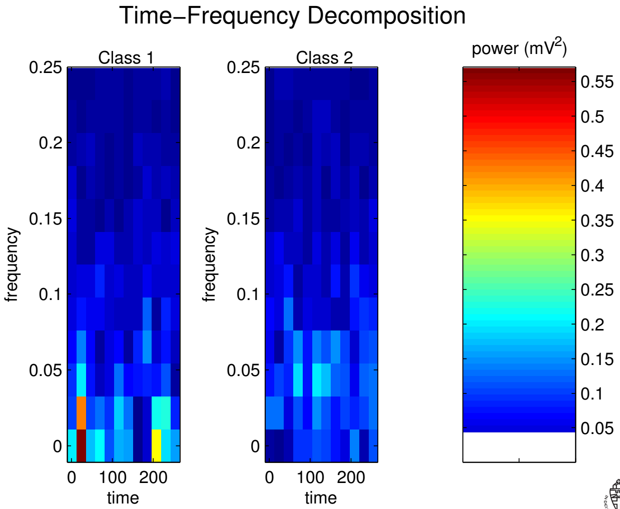
- Short-time-Fourier-transform (STFT)
- Hilbert Transform
- Multi-Taper-Method
- Wavelet-Transform

All essentially the same and only talk about the STFT here.

Time-Frequency Decomposition - easy



Time-Frequency Decomposition - sim data

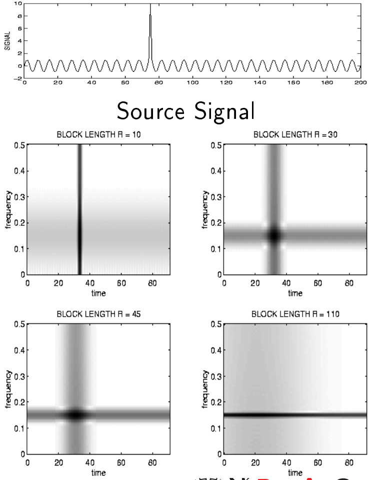


Time-Frequency trade-offs

Time-Frequency trade-off

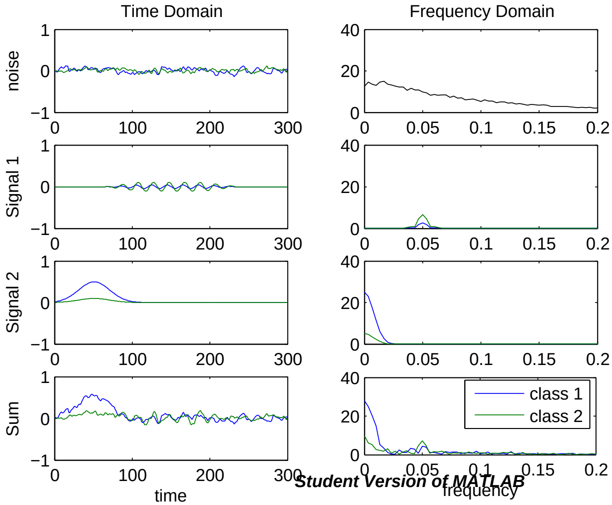
Time-Resolution (ΔT) and Frequency-Resolution (ΔF) are inversely related to each other by: $\Delta f = 1/\Delta T$.

- Increasing time resolution decrease frequency resolution
- Increasing frequency resolution decreases time resolution
- Intuitively $\rightarrow$ need $>$1 cycle to reliably identify phase+amplitude

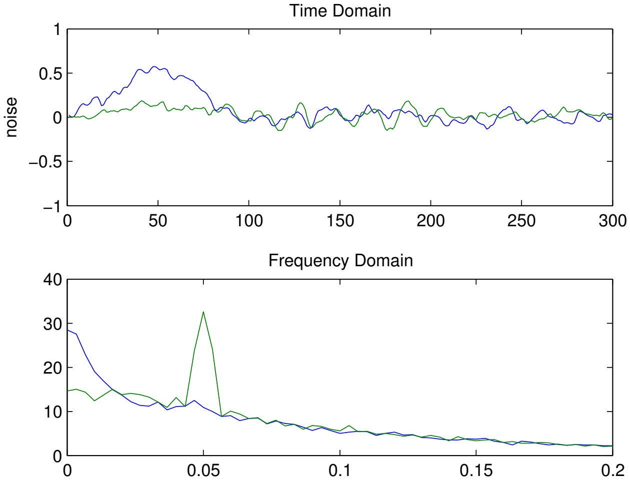


Time Frequency Decomposition

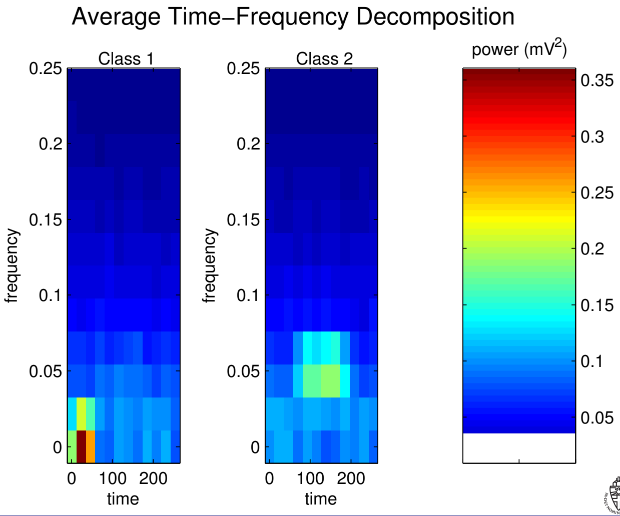
Averaging signals



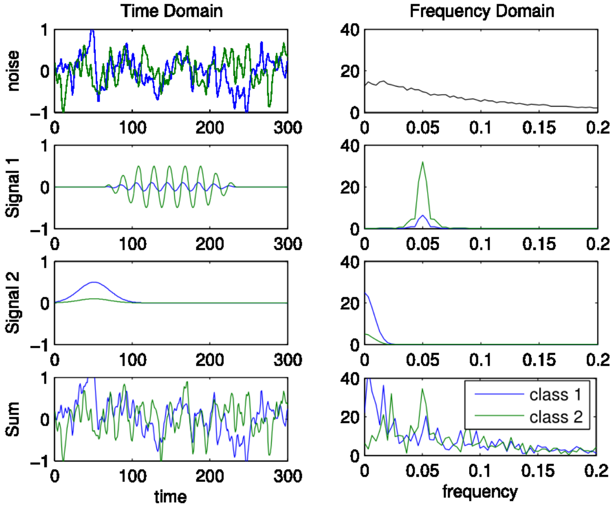
Averaging: Time-domain vs. Freq-domain



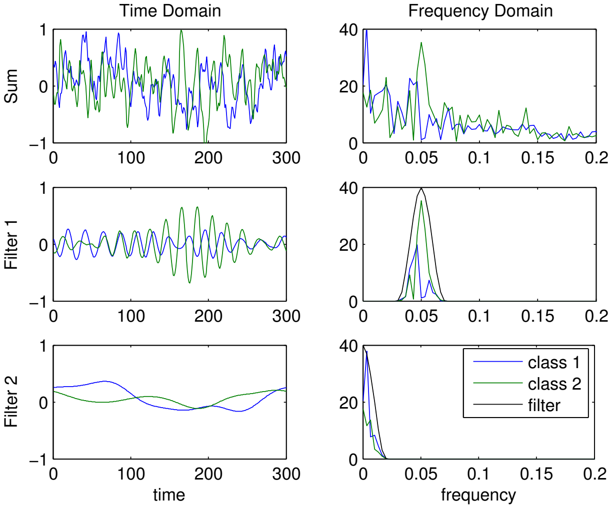
Averaging: Time-Frequency-Domain (sim-data)



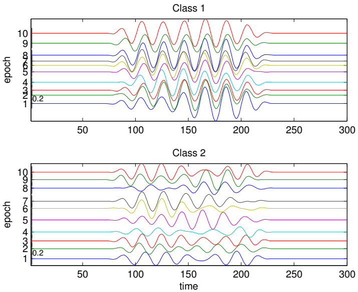
Spectral-filtering



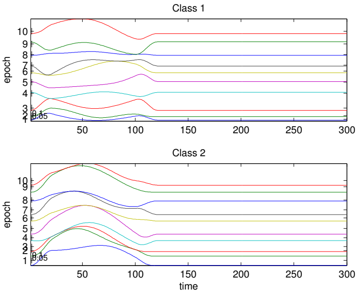
Spectral-filtering (2)



Spectral-filtering



Spectral filter 1



Spectral filter 2

Summary signal filtering

- signal filtering is used to attenuate background noise...
- ... and hence make the target signal easier to detect

Most commonly used filters:

1. temporal – focus 'when' signal is expected
2. spectral – focus on frequency signal is expected to have
3. spatial – focus on where signal is expected to come from
4. averaging – special type of temporal which attenuates uncorrelated noise.

Note:

To use design optimal filters need to know the expected characteristics of the target signal (and noise).

Learning Objectives

- Understand why pre-processing is necessary in BCI
- Describe the main sources of artifacts and their effects on the recorded signal; i.e. muscles, eyes, movement, hardware, line-noise etc.
- Describe the main types of pre-processing and what they are useful for; re-referencing, spectral-filtering, artifact-detection/rejection/removal
- Understand what is shown in the temporal, spectral and time-frequency representations of a signal
- Understand how signal filtering in these different representations can be used to suppress the noise in measured data, and hence make a signal more visible
- Understand how knowledge of the signal characteristics allows one to design signal filters to improve signal-to-noise