The Artifact Subspace Reconstruction Method

Christian A Kothe

intheon

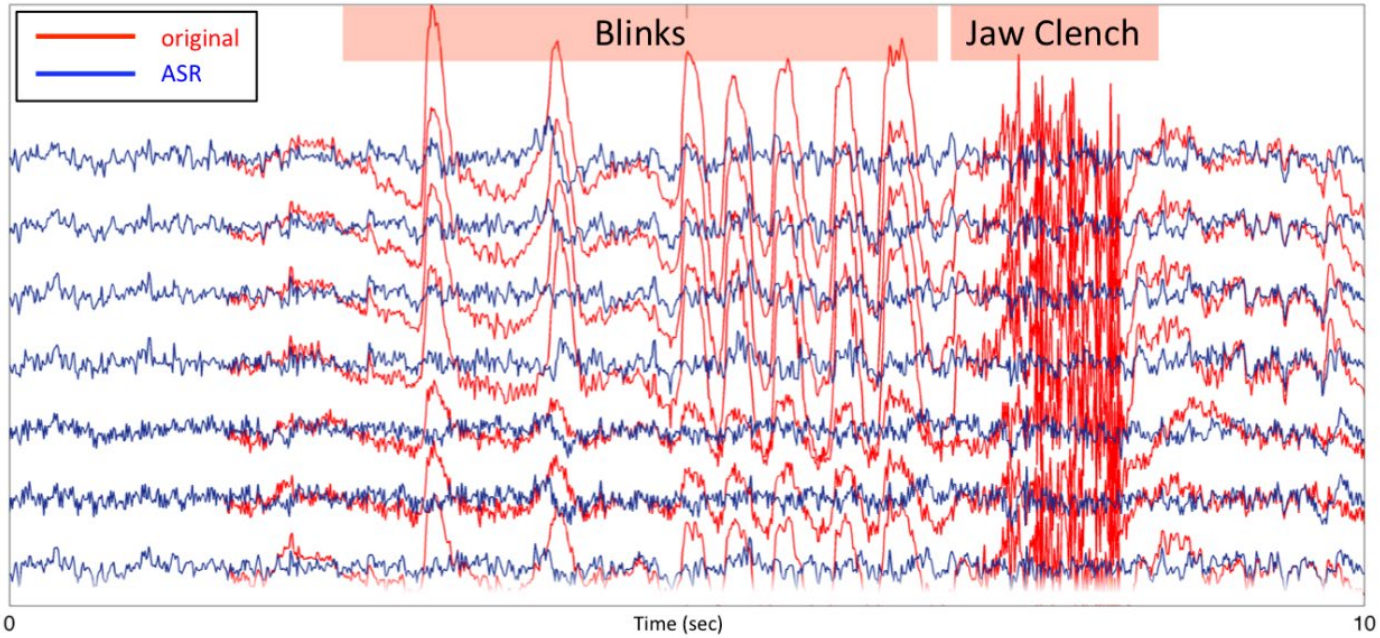
Background

- Originally created at SCCN in 2013
- Developed as part of Cognition and Neuroergonimics CTA program to support ambulatory neuroscience/-ergonomics research & applications

Funded by ARL under Cooperative Agreement Number W911NF-10-2-0022

Artifact Subspace Reconstruction

- Method for removing high-amplitude artifacts from EEG

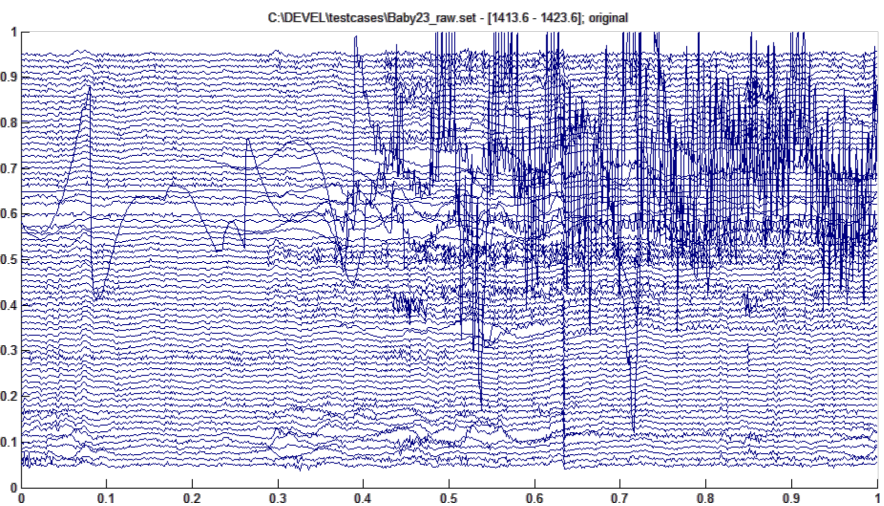


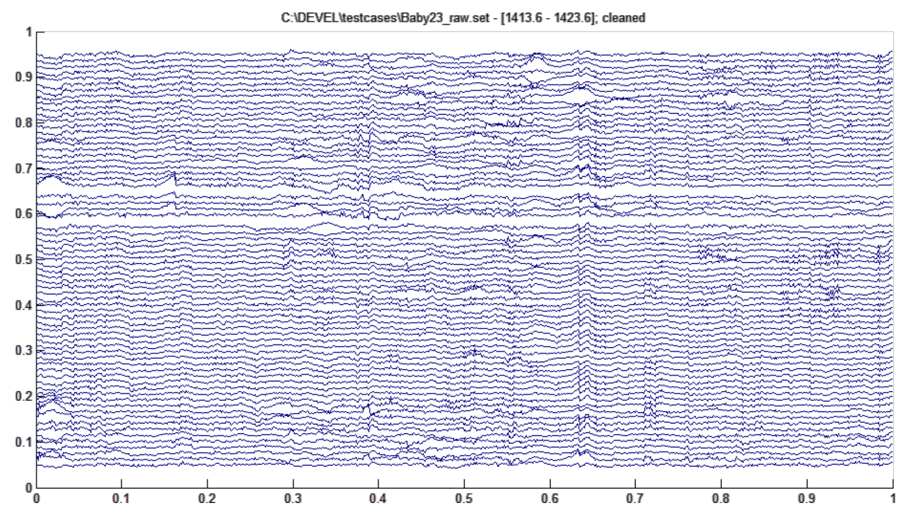
- Reconstructs underlying EEG activity, recovers missing energy, leaves no pathologies in signal

visualization generated by vis_artifacts

Artifact Subspace Reconstruction

- Designed to handle non-stationary / non-repeating artifacts



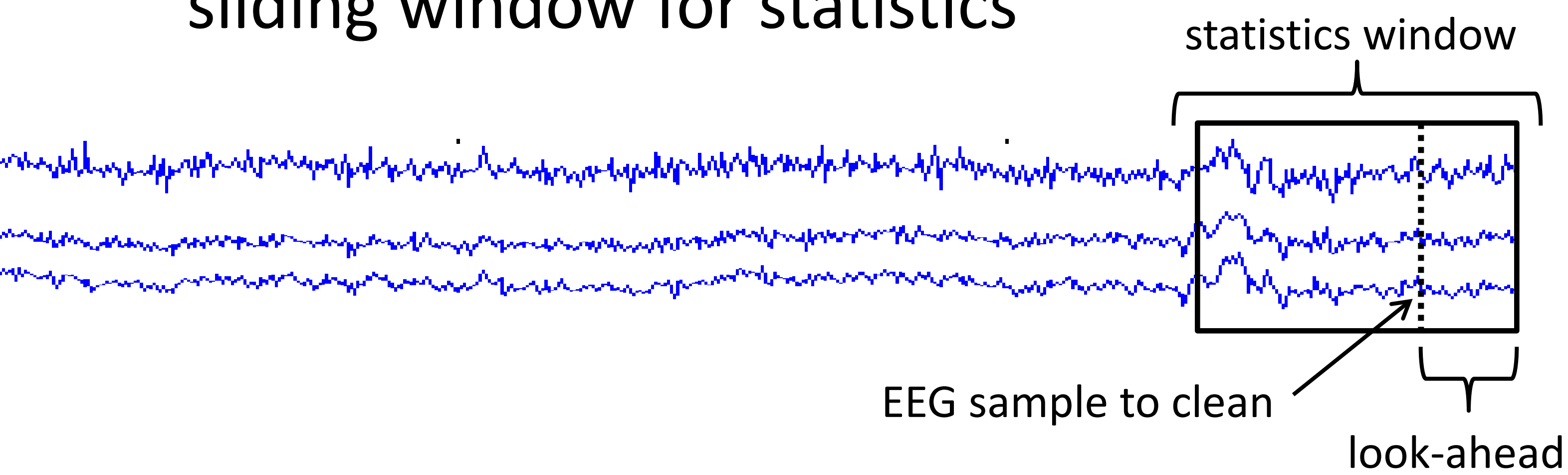


Prerequisites

- ASR requires data to be already high-pass filtered (signal mean should be *zero*)
- Data is assumed to be *full rank* (i.e., cannot use common average reference before ASR)
- Best to remove misc channels beforehand (e.g., trigger channel, packet counter, esp. flatline channels)
- Keeping EXG channels *should* work, but better to test with and without

Sliding-Window Processing

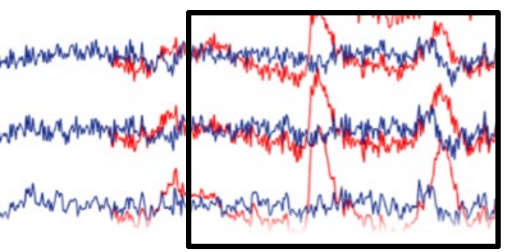
- Applied independently sample-by-sample, using a sliding window for statistics



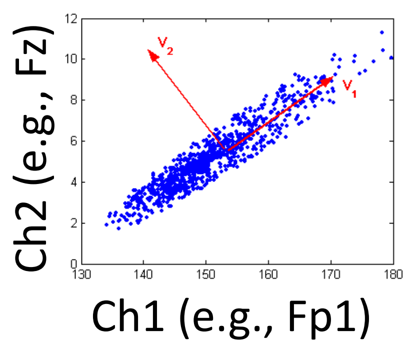
- Sliding window is typically short (to handle non-stationary / non-stereotypical artifacts), but longer windows can be used

Finding Artifact Components

- Want to find only *high-amplitude* signal components (potential artifact components) in the short statistics window
- Can be done using Principal Component Analysis (PCA) on signal in statistics window **W** [nCh x nWnd]:

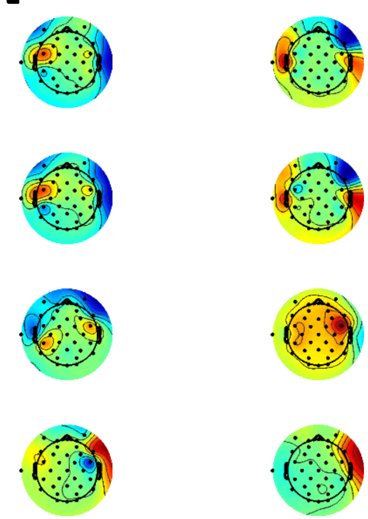


Raw Signal Window W



PCA Components

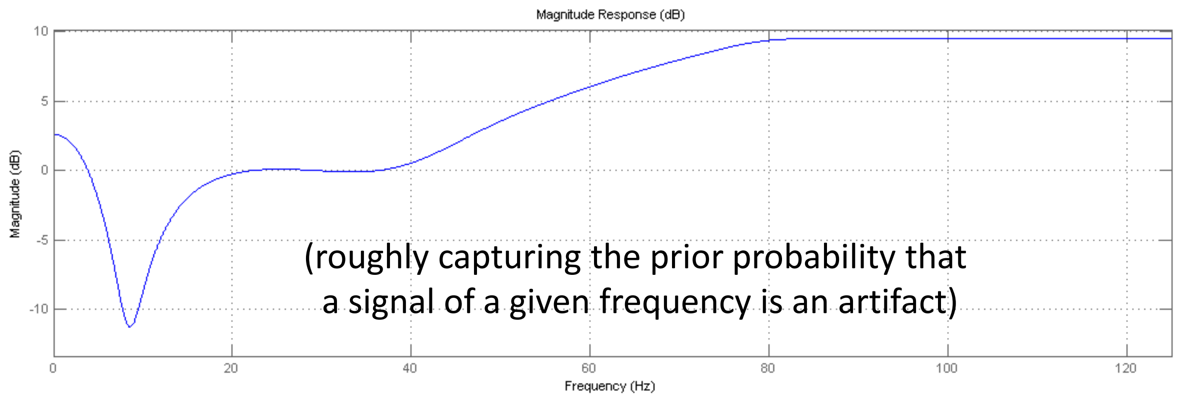
$$V = \mathbf{eig}(WW^\top)$$



Random sample of high-variance components

Optional Spectral Weighting

- Statistics can be *spectrally weighted* by applying an IIR filter to the data in the statistics window (8-ord. Yule-Walker), used by default



- This filter is rarely touched, but can be used to make ASR “blind” to certain desired signals (e.g., SSVEP)
- Can also be used to cause ASR to be more sensitive to certain kinds of artifacts (e.g., muscle, EM)

Artifact Detection Threshold

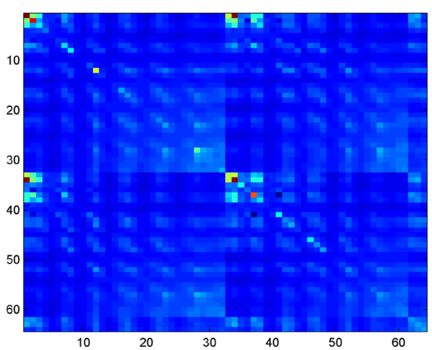
- Applied to the variance for each detected component
- Needs to separate artifacts from non-artifacts (i.e., background EEG, ERPs)
  →Threshold is data-dependent
- Different brain areas will have different background activity levels (e.g., occipital alpha)
  →Threshold is direction-dependent
- Need statistics of “clean” EEG data to calibrate threshold

Finding Clean Calibration Data

- For real-time use, ASR must be calibrated on some initial reasonably clean resting-state data (can be eyes-closed), e.g. 1 minute
- On recorded data, ASR will optionally use clean_windows to find the cleanest data segments in a recording (e.g., 10% / few minutes), on by default
- Fine if these data *still* have artifacts in them → usually don’t have to worry about it

Clean-Data Statistics

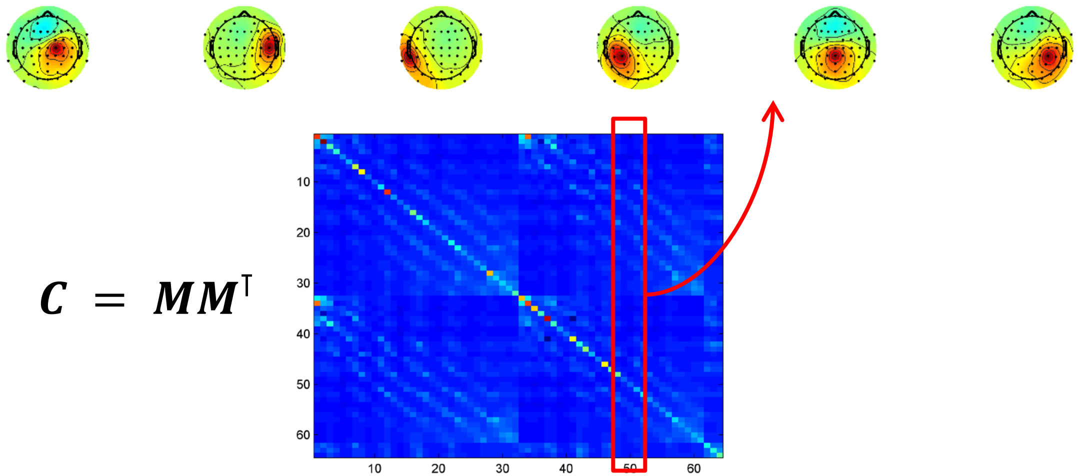
- We extract several statistics from clean data D:
  - Robust covariance matrix
  - Robust mixing matrix
  - Direction-dependent threshold matrix



Covariance Matrix C

$$C = DD^{\top}/nT$$

$$C = MM^{\top}$$



Mixing Matrix M

Robust Estimators

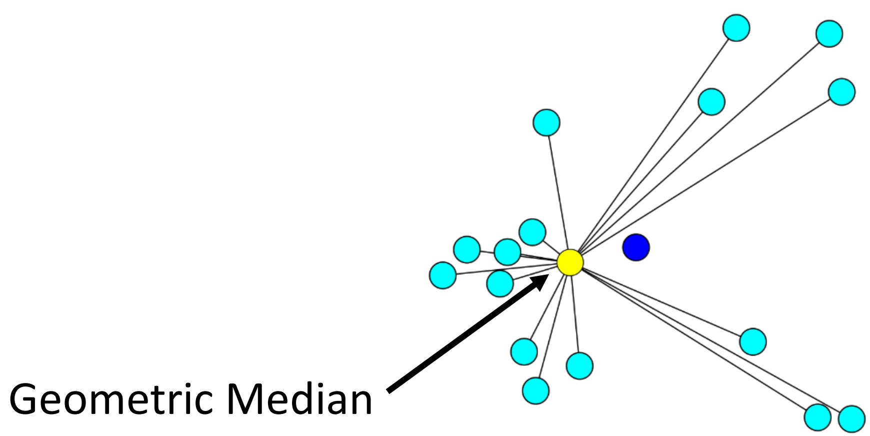
- We use a robust estimator to estimate the covariance matrix (geometric median)

$$\boldsymbol{\mathcal{G}}(\boldsymbol{X}) = \arg\min_{\boldsymbol{y}} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{y}\|_2$$

Geometric Median

$$y_{i+1} = \left(\sum_{j=1}^{m} \frac{x_j}{\|x_j - y_i\|}\right) \Bigg/ \left(\sum_{j=1}^{m} \frac{1}{\|x_j - y_i\|}\right)$$

Iterative Formula
(iteratively reweighted least squares)



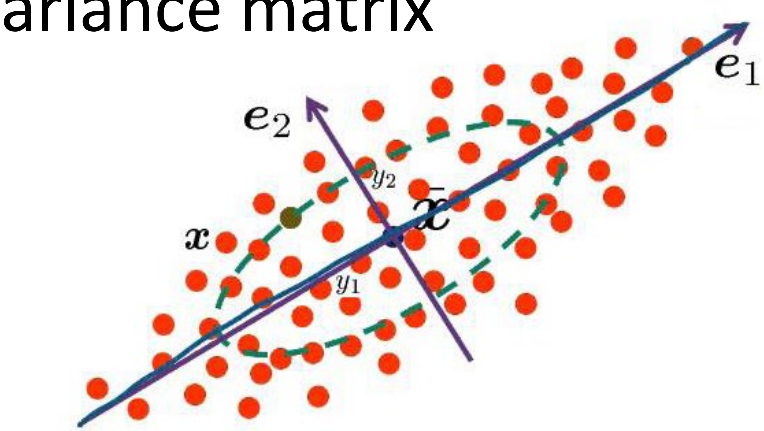
$\boldsymbol{x}_i$ is a (vectorized) covariance matrix of a short window of ***D***

Detection Threshold

- Artifacts components detected using a threshold function $\mathrm{th}(v_k)$
- Can be thought of as a robust Mahalanobis distance estimated from clean data *D*

$$\mathrm{th}(v_k) \coloneqq \| diag(z)\bar{P}^T v_k \|$$

- $\bar{P}$ are the principal components derived from the robust covariance matrix

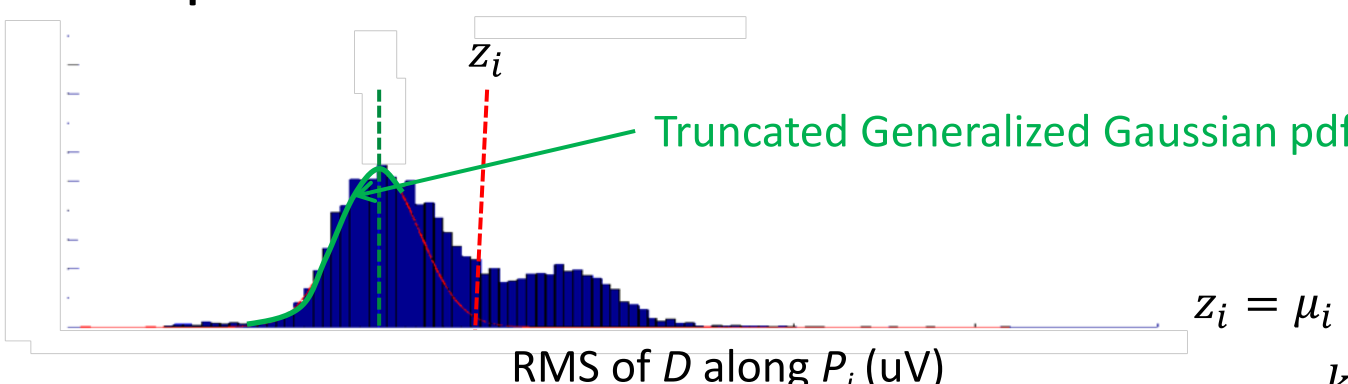


Detection Threshold

- Artifacts components detected using a threshold function $\mathrm{th}(v_k)$
- Can be thought of as a robust Mahalanobis distance estimated from clean data *D*

$$\mathrm{th}(v_k) \coloneqq \| diag(z)\bar{P}^T v_k \|$$

- z_i are the thresholds along each clean data component's direction



$$z_i = \mu_i + \sigma_i k$$

k → **cutoff parameter**

Rejection Rule

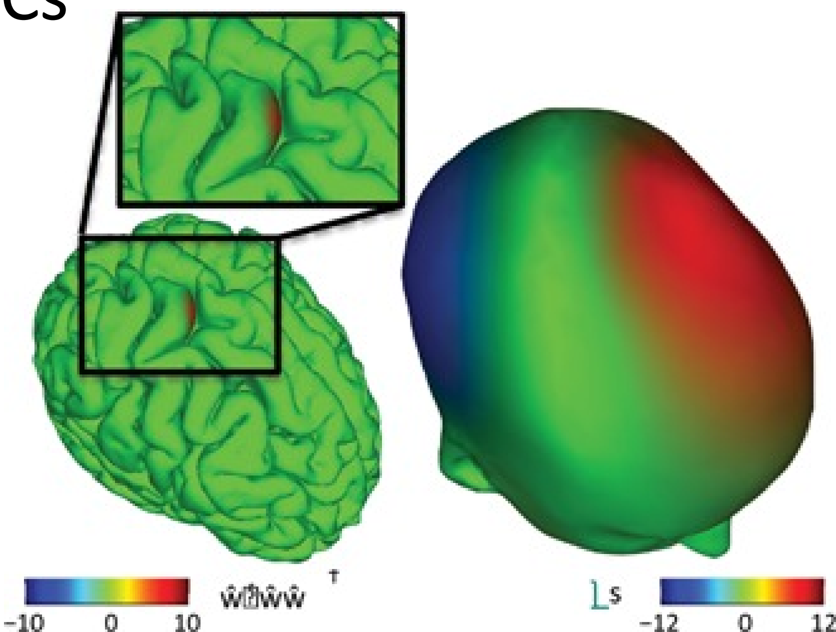
- Reject if short-window PCA component's standard deviation exceeds threshold along its direction

$$\sqrt{\lambda_k} = \sigma_k > \mathrm{th}(v_k)$$

- Now what do we do with rejected components?

Reconstructing Rejected Components

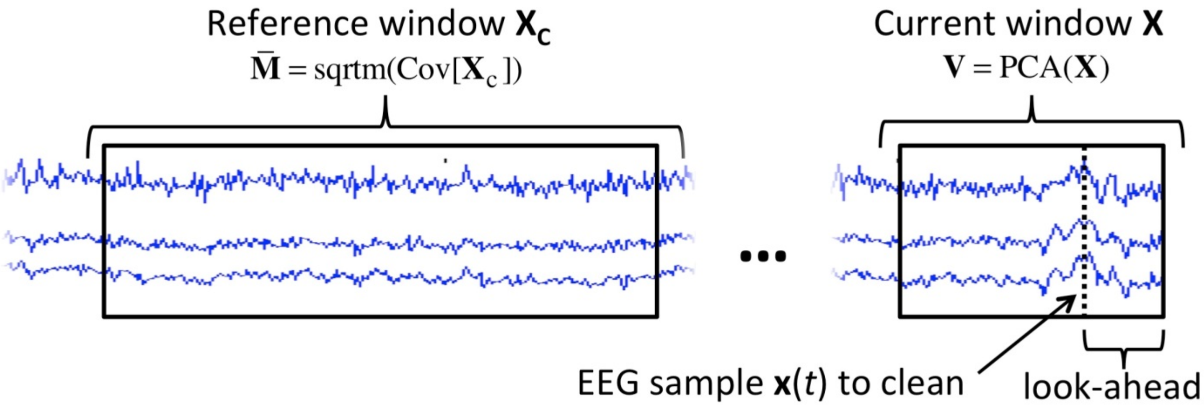
- Any two EEG channels are *almost always* correlated
- So are any two linear combinations of channels
- So are the components in our short statistics window
- Background EEG content in artifact PCs can be estimated from non-artifact PCs
- **Analogy:** estimating one channel from other channels
- Here not using channels but artifact / non-artifact subspaces on a short window



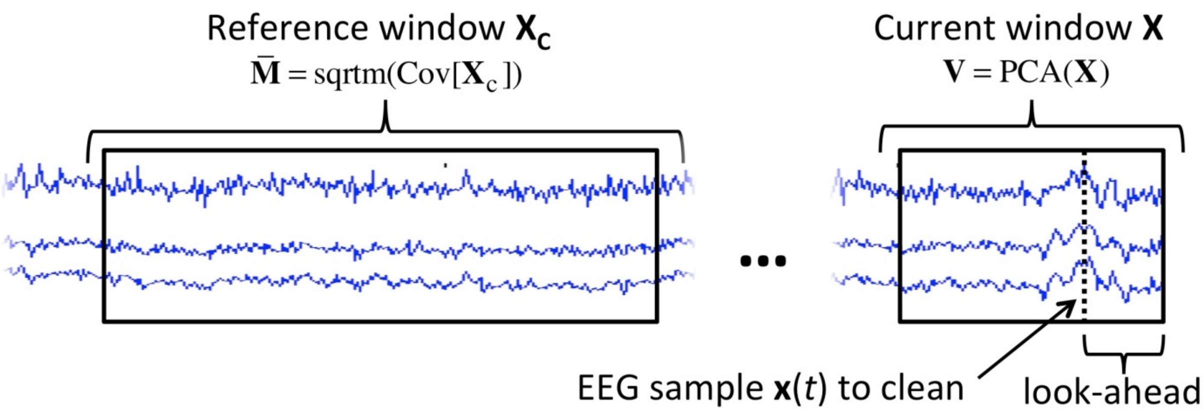
Reconstructing Rejected Components

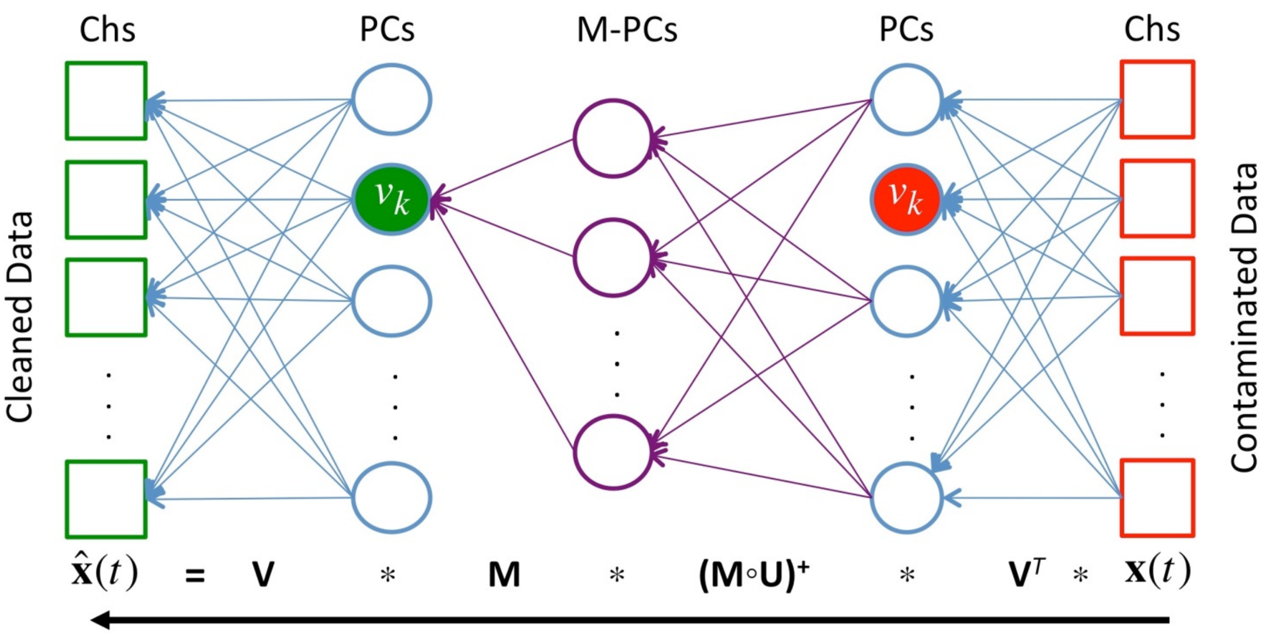
- But isn't the artifact subspace uncorrelated from the non-artifact subspace?
- Only in the short-window decomposition
- Not in the average background EEG statistics (correlation structure)
- This correlation structure is described by our robust mixing matrix *M* (from clean data)

Performing the Reconstruction



Performing the Reconstruction





$$M = V^T \bar{M}$$

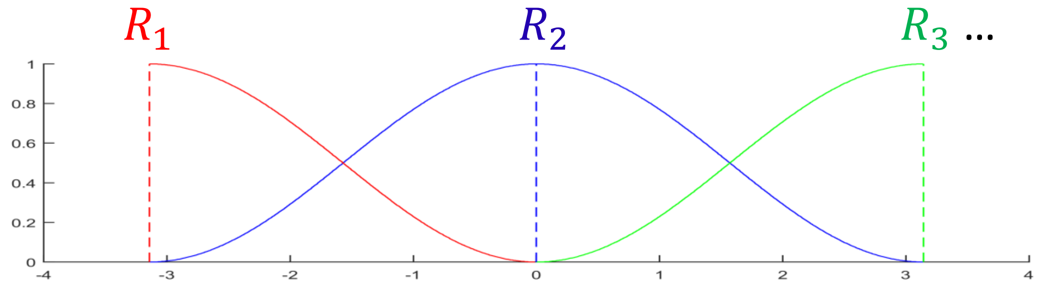
$$U_{kl} = \begin{cases} 0, & \sigma_k > \mathrm{th}(v_k) \\ 1, & \text{otherwise} \end{cases}$$

$$R = VM(M \circ U)^+ V^T$$

$$\hat{x}(t) = Rx(t)$$

Overlapped-Window Composition

- Re-calculate R for every k'th sample (step size) (in real time can reconstruct at end of each chunk)
- Blend between successive R estimates for intermediate samples, e.g., using raised-cosine weighting



- Avoids sharp transients (jump discontinuities)
- Also maintains full-rank output data (i.e., no pathological subspaces)

Future Directions

- Calibrate in real time on expanding window or sliding window (difficult due to truncated generalized Gaussian est.)
- Use Riemannian estimator for calibration covariance
- Find good rules of thumb for threshold

Thanks!

Questions?

Figure References

- Brain model: Zeynep Akalin Acar
- Geometric median figure: Wikipedia
- Demo video: Cognionics