Assignment 2

Mahdi Tabatabaei - Heliya Shakeri

Biomedical Signal and Image Processing Lab

Nastaran Nouri

October 16, 2024

Contents

Noise Removal in Simulated Non-Seizure EEG Signals

In this section, we investigate noise removal using the ICA method applied to simulated EEG signals. We use the provided dataset of clean (desired) and noisy EEG signals. The aim is to compare signals with different SNR levels and remove noise sources to reconstruct the clean signals using ICA.

Question 1

Plot the clean EEG signal `X_org` and the noisy EEG signal `X_noise`. Use the `disp_eeg` function to visualize the channels and set the channel labels and time axis accordingly.

```
% Load the original EEG data
load('X_org.mat');
load('X_noise.mat');
load('Electrodes.mat');
Fs = 250;

% Plot the clean EEG signal
offset = max(max(abs(X_org))) / 1.5;
feq = Fs;
ElecName = Electrodes.labels;
disp_eeg(X_org, offset, feq, ElecName);
title('Clean Data');

% Plot the noisy EEG signal
offset = max(max(abs(X_noise))) / 1.5;
feq = Fs;
ElecName = Electrodes.labels;
disp_eeg(X_noise, offset, feq, ElecName);
title('Noisy Data');
```

Source Code 1: EEG - Question 1: Clean and Noisy Signal Plot

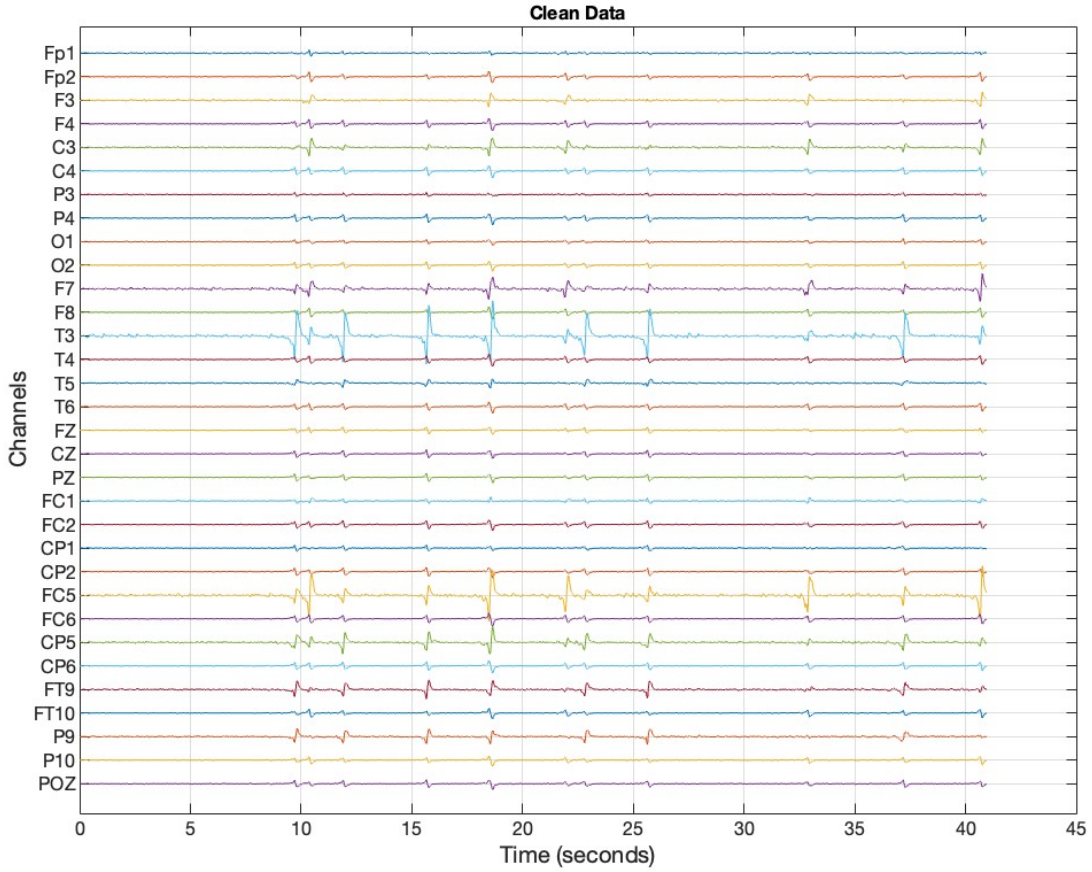

Figure 1: Clean EEG Signal

Question 2

Plot `X_noise` and identify the noise artifacts. Label the channels and time on the plot.

```
% Plot the noisy EEG signal with labels
offset = max(max(abs(X_noise))) / 1.5;
feq = Fs;
ElecName = Electrodes.labels;
disp_eeg(X_noise, offset, feq, ElecName);
title('Noisy EEG Data');
```

Source Code 2: EEG - Question 2: Noisy EEG Signal

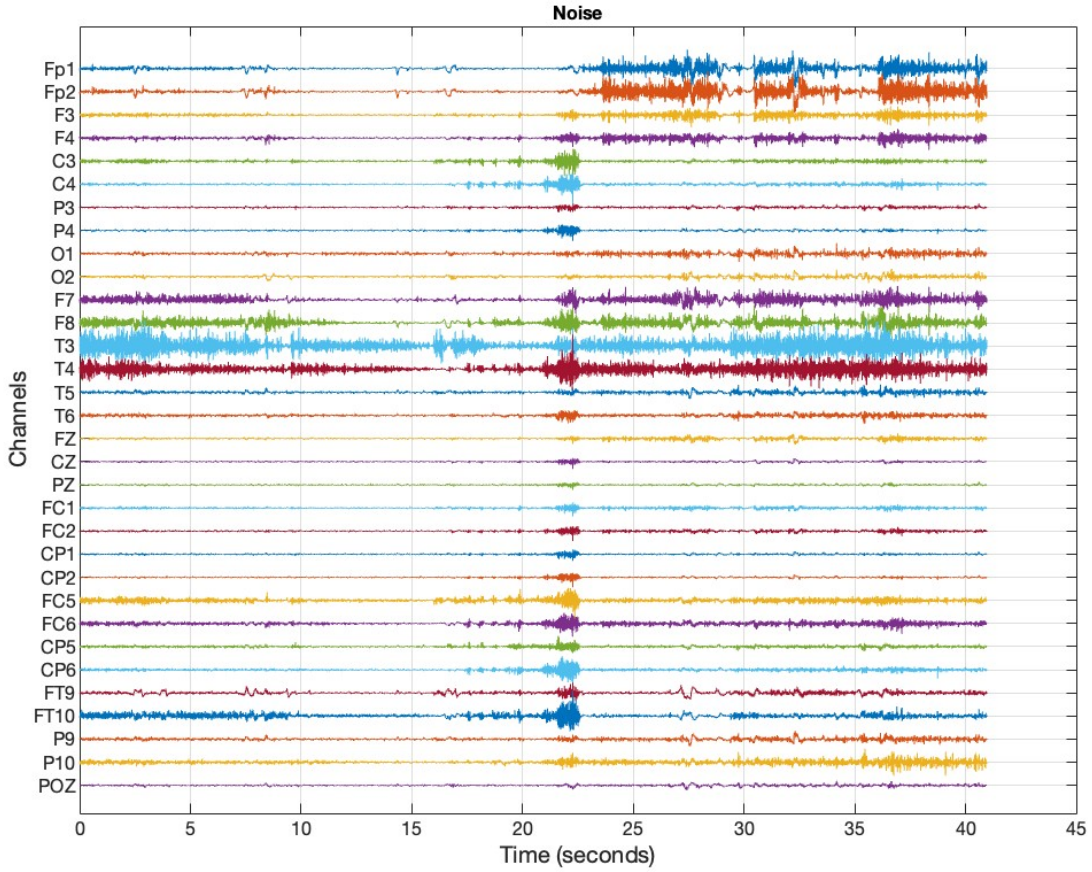

Figure 2: Noisy EEG Signal

Question 3

Plot the EEG signals with different SNRs (-5 dB, -15 dB) combined with the main signal. Plot the signals with `SNR = -15 dB` and compare them to the clean signal without noise.

```
%Calculate energy of the original and noise signals
cal_energy = @(signal) sum(sum(signal.^2));
Ps = cal_energy(X_org);
Pn = cal_energy(X_noise);

% SNR values in dB
SNR = [-5, -15];
noisy_sig = {};

% Loop through each SNR value
for iSNR = 1:length(SNR)
    sigma = sqrt(Ps/Pn * 10.^(-SNR(iSNR)/10));
    noisy_sig{iSNR} = X_org + sigma * X_noise;
end

% Plot the noisy signal with the lowest SNR
offset = max(max(abs(noisy_sig{2})))/2;
feq = Fs;
ElecName = Electrodes.labels;
disp_eeg(noisy_sig{2}, offset, feq, ElecName);
title('Noisy EEG Signal with SNR = -15 dB');
```

Source Code 3: EEG - Question 3: Signals with Different SNRs

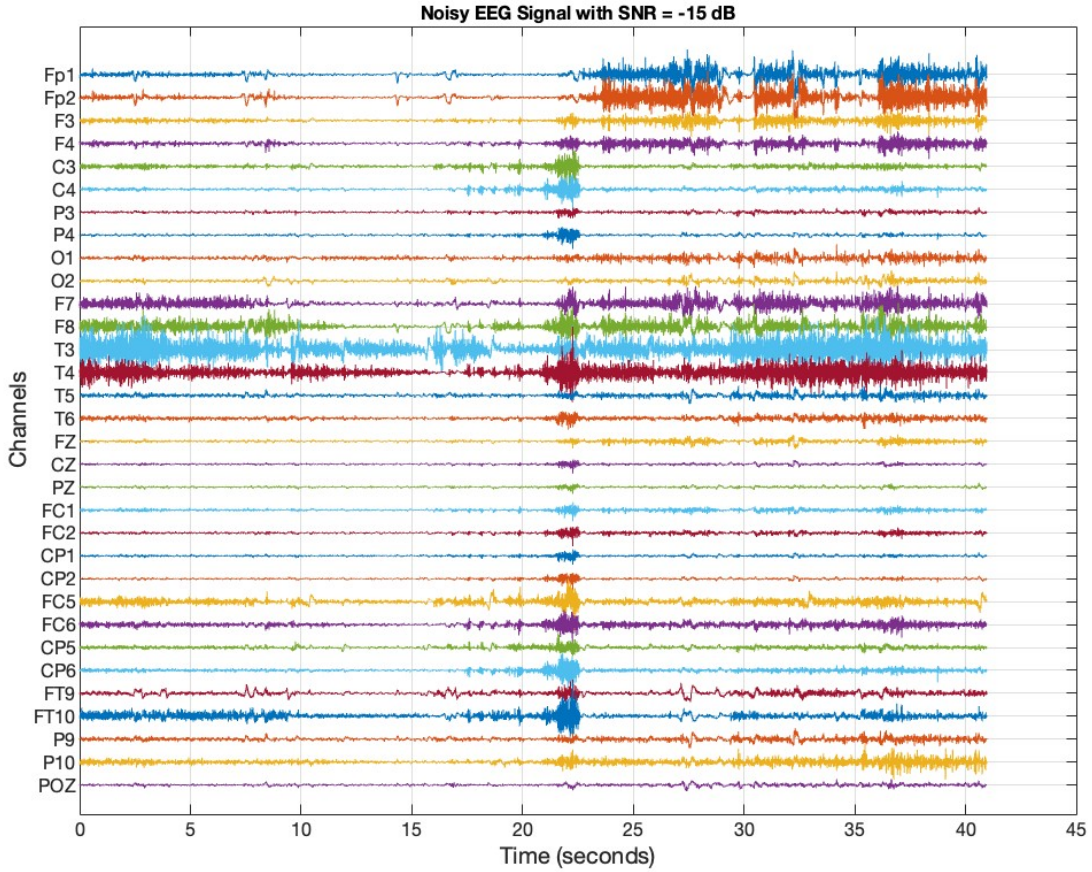

Figure 3: EEG Signal with SNR = -15 dB

Question 4

Use the ICA method to extract the desired sources. You can use the `COM2R.m` algorithm or any other ICA method.

```
%noisy_signal = noisy_sig{1};
noisy_signal = noisy_sig{2};

% Estimate the number of independent sources
Pest = size(noisy_signal,1);

% Apply the COM2R algorithm to the noisy signal
[F, W, K] = COM2R(noisy_signal,Pest);

% Extract the estimated sources
Z = W * noisy_signal;
```

Source Code 4: EEG - Question 4: ICA Source Extraction

Question 5

Review all extracted sources and remove unnecessary ones (e.g., noise sources) while keeping desired sources.

```
% Plot all the extracted sources
disp_eeg(Z, offset/0.5, feq, []);
title('Extracted EEG Sources from ICA');
```

Source Code 5: EEG - Question 5: Source Review and Noise Removal

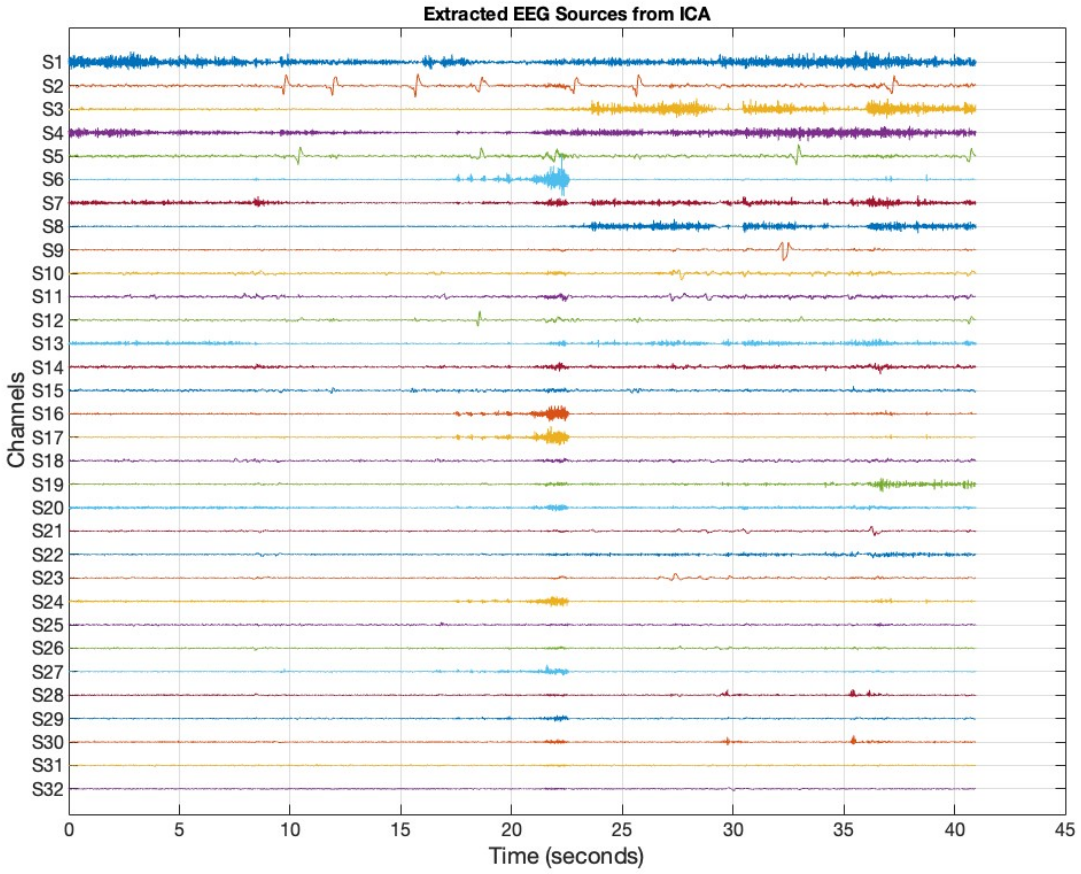

Figure 4: Cleaned EEG after Source Selection: SNR = -5dB

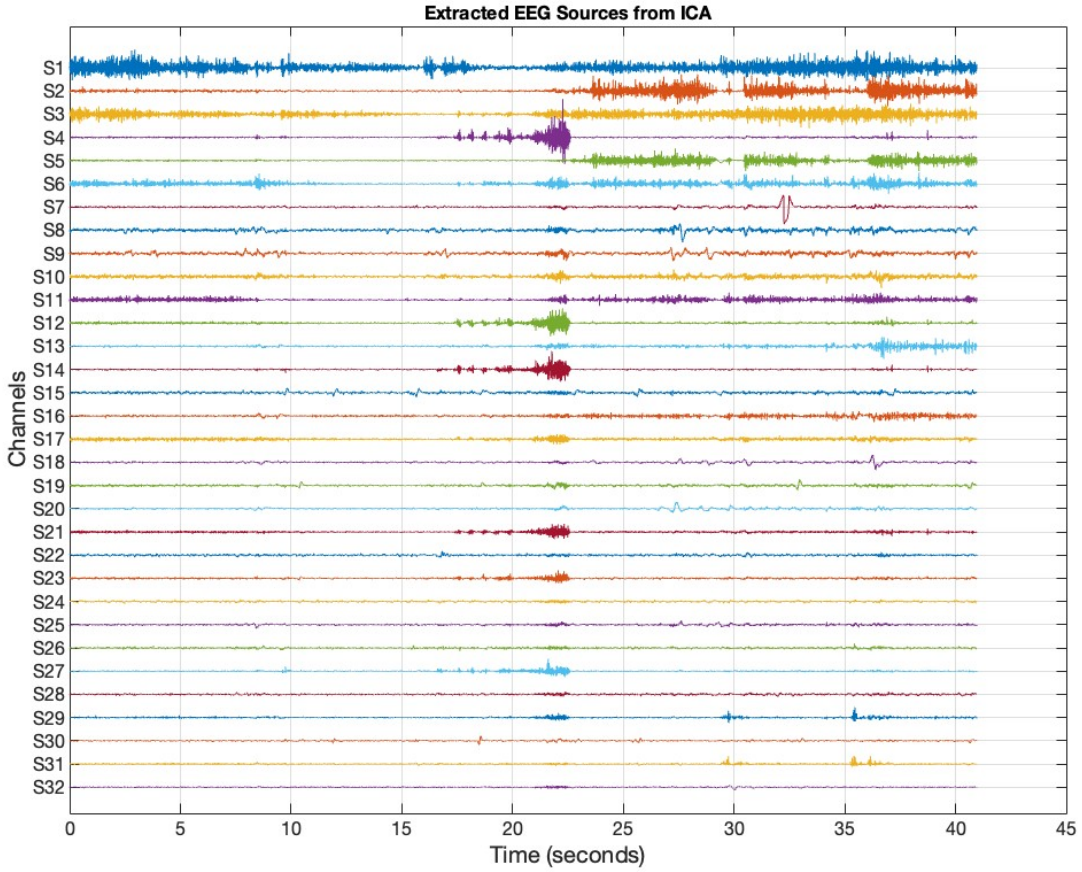

Figure 5: Cleaned EEG after Source Selection: SNR = -15dB

Question 6

Return the desired sources to the sensor domain (observation domain) and reconstruct the EEG signal with noise removed.

```
% Select the components
%keeping_comp = [2, 5, 10, 12, 21, 23];
keeping_comp = [15, 18, 19];

Z_keep = Z(keeping_comp, :);
F_keep = F(:, keeping_comp);

% Reconstruct the denoised EEG signal
X_Den = F_keep * Z_keep;

% Plot the denoised EEG signal
offset = max(max(abs(X_Den)))/1.5;
feq = Fs;
ElecName = Electrodes.labels;
disp_eeg(X_Den,offset,feq,ElecName);
title('Denoised EEG Signal after ICA Source Selection');
```

Source Code 6: EEG - Question 6: Reconstructed EEG Signal

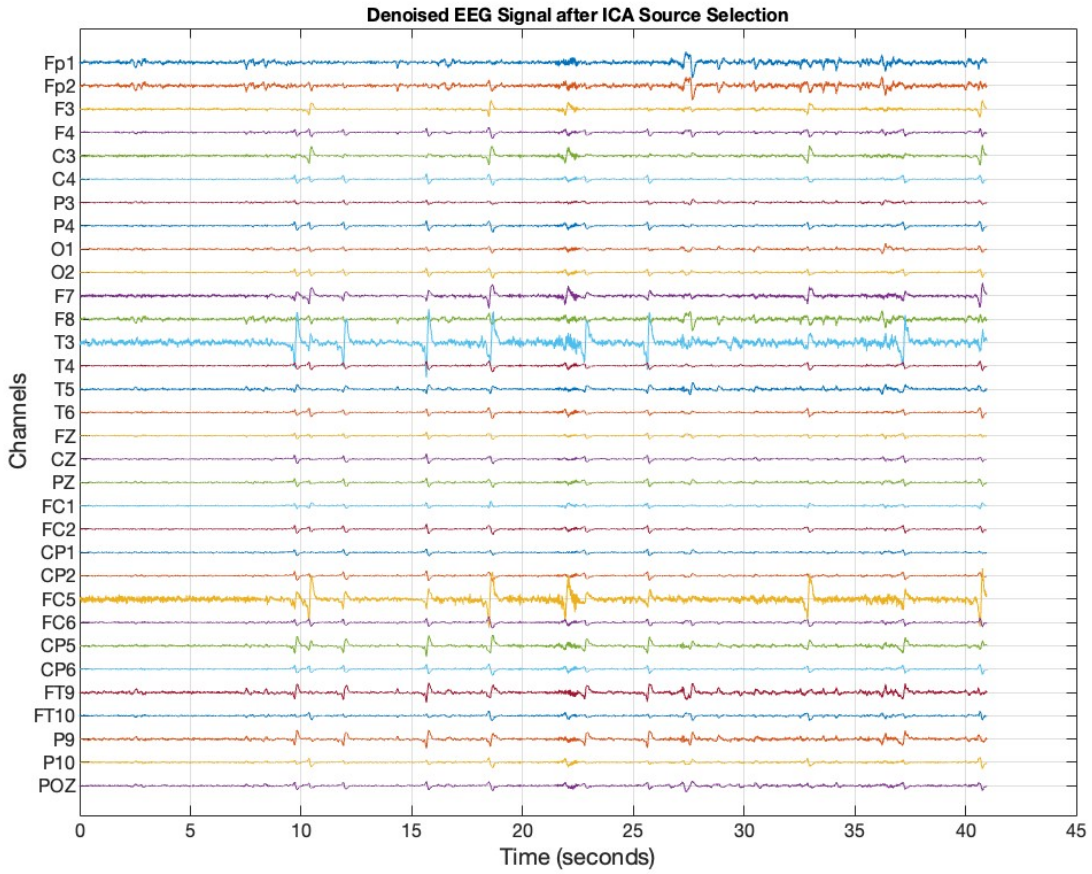

Figure 6: Reconstructed EEG Signal after Noise Removal: SNR = -5dB

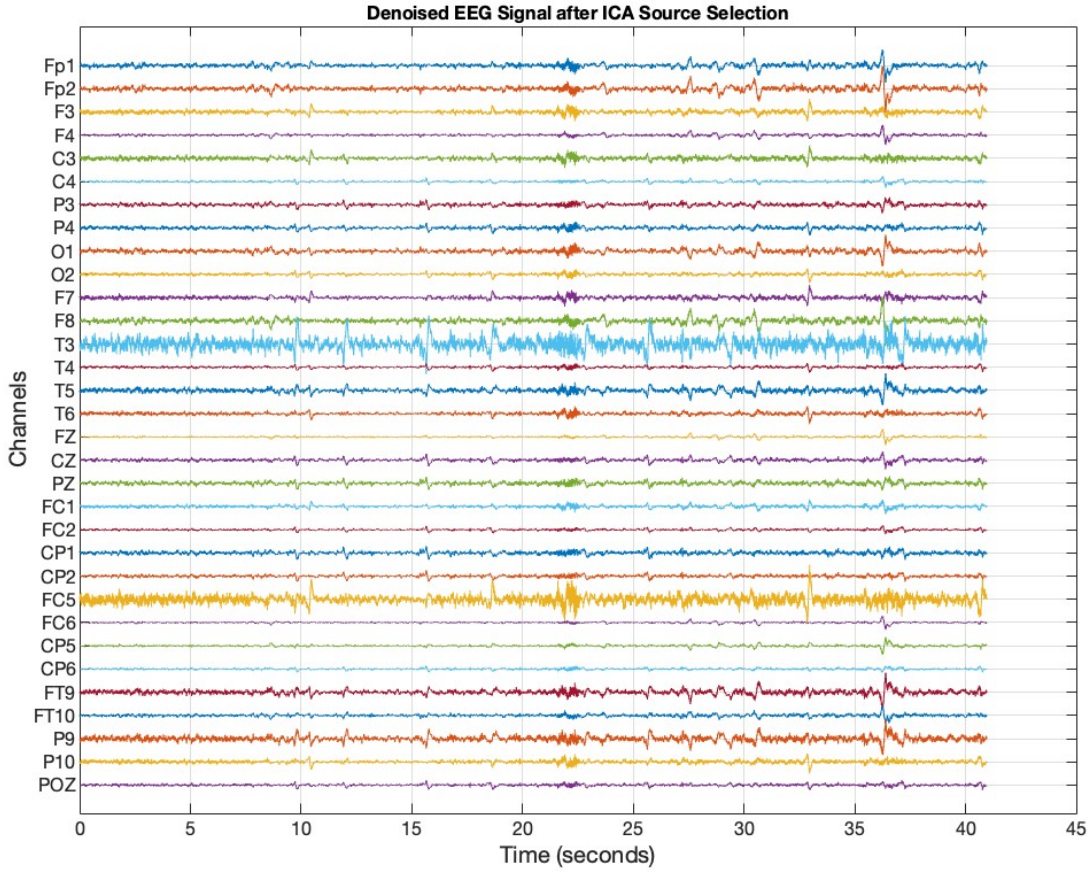

Figure 7: Reconstructed EEG Signal after Noise Removal: SNR = -15dB

Question 7

Compare the noise-removed signal for channels 13 and 24 with the original and noisy signals. Plot and analyze the results.

```
% Time vector
t = (0:size(X_org, 2) - 1) ./ Fs;

% Plot Channel 13
figure;
subplot(2, 1, 1);
plot(t, X_Den(13, :), 'LineWidth', 1.5);
hold on;
plot(t, X_org(13, :), 'LineWidth', 1.5);
ylim([-50 50]);
xlabel('Time(s)');
ylabel('Amplitude');
title('Channel 13: Denoised vs Original');
grid on; grid minor;
xlim([t(1) t(end)]);
legend('X Den', 'X Org');

% Plot Channel 24
subplot(2, 1, 2);
plot(t, X_Den(24, :), 'LineWidth', 1.5);
hold on;
```

```
    plot(t, X_org(24, :), 'LineWidth', 1.5);
    ylim([-50 50]);
    xlabel('Time(s)');
    ylabel('Amplitude');
    title('Channel 24: Denoised vs Original');
    grid on; grid minor;
    xlim([t(1) t(end)]);
    legend('X Den', 'X Org');
```

Source Code 7: EEG - Question 7: Channel Comparison

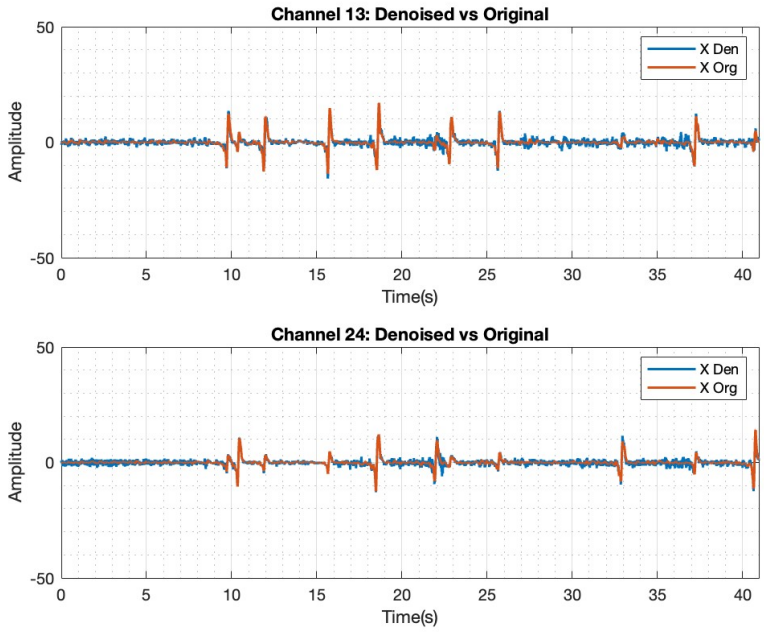

Figure 8: Channels 13 and 24, Comparison of Original and Denoised Signals: SNR = -5dB

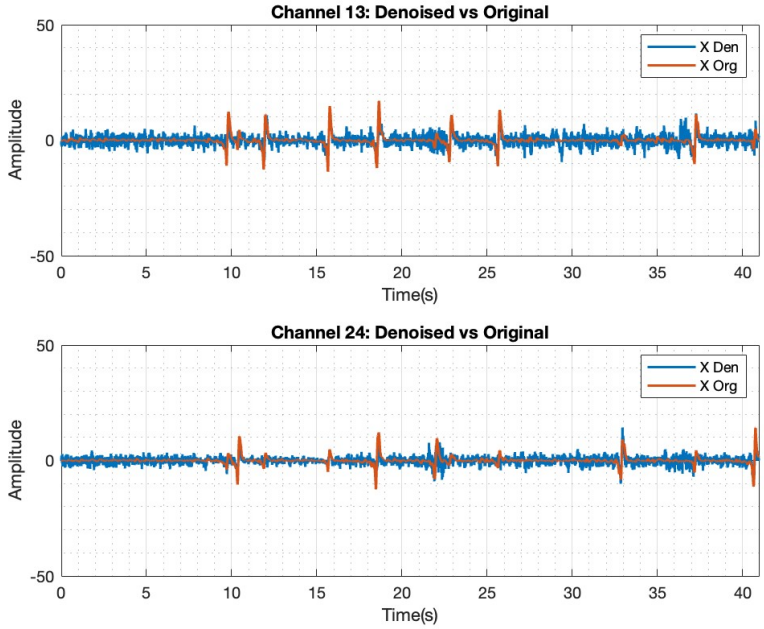

Figure 9: Channels 13 and 24, Comparison of Original and Denoised Signals : SNR = -15dB

Question 8

Calculate the Relative Root Mean Square Error (RRMSE) for each SNR level and analyze the results.

The Relative Root Mean Square Error (RRMSE) is calculated using the following equation:

$$\text{RRMSE} = \frac{\sqrt{\sum_{n=1}^{32}\sum_{t=1}^{T}\left(x_{\text{org}}^{(n)}(t) - x_{\text{den}}^{(n)}(t)\right)^2}}{\sqrt{\sum_{n=1}^{32}\sum_{t=1}^{T}\left(x_{\text{org}}^{(n)}(t)\right)^2}}$$

```
% Number of channels and time points
num_channels = size(X_org, 1);
num_timepoints = size(X_org, 2);

% Calculate the numerator
squared_error_sum = sum((X_org - X_Den).^2, 'all');

% Calculate the denominator
original_signal_energy = sum(X_org.^2, 'all');

% Compute RRMSE
rrmse = sqrt(squared_error_sum / original_signal_energy);
disp(['Relative Root Mean Square Error (RRMSE): ', num2str(rrmse)]);
```

Source Code 8: EEG - Question 8: RRMSE Calculation

Soloution

The calculated Relative Root Mean Square Error (RRMSE) values for the two SNR levels are:

- SNR = -5 dB

  Relative Root Mean Square Error (RRMSE): 0.61133

- SNR = -15 dB

  Relative Root Mean Square Error (RRMSE): 1.2117

Analysis:
The comparison of RRMSE values for different SNR levels helps in understanding the effectiveness of the noise removal process.

- **SNR -5 dB (RRMSE = 0.61133)**: The lower RRMSE value at -5 dB shows that the noise removal process is much more effective in this case. With less noise in the signal, the algorithm can reconstruct the clean signal with much greater accuracy, resulting in a smaller error.
- **SNR -15 dB (RRMSE = 1.2117)**: This higher RRMSE value indicates that the reconstructed signal deviates significantly from the original clean signal. The large error reflects the difficulty in accurately removing noise at a lower SNR where the signal is highly corrupted by noise.

Conclusion:
The RRMSE values demonstrate that the quality of signal reconstruction is heavily dependent on the level of noise present in the original signal. At higher noise levels (SNR = -15 dB), the denoised signal has more errors, while at lower noise levels (SNR = -5 dB), the signal can be reconstructed more accurately. This highlights the challenges noise removal algorithms face when dealing with highly noisy data and how cleaner input signals yield better denoising results.

Noise removal of real epileptic signals

The folder related to this experiment (Lab_2) contains four .mat files. These files include EEG signals: (NewData1) pertains to rapid eye movement in patients, and (NewData4) contains signals related to non-seizure intervals in epilepsy patients. The purpose of the experiment is to remove noise and artifacts from the signals, using the ICA method to extract two signals from the four channels. Follow the steps below regarding each signal, and obtain the results.

Question 1

Plot the signal in the time domain and label all the channels.

```
offset = max(max(abs(signal)))/1;
feq = Fs;
disp_eeg(signal,offset,feq,labels);
xlim("tight")
title("Signal 1")
```

Source Code 9: Question 2 - Part 1

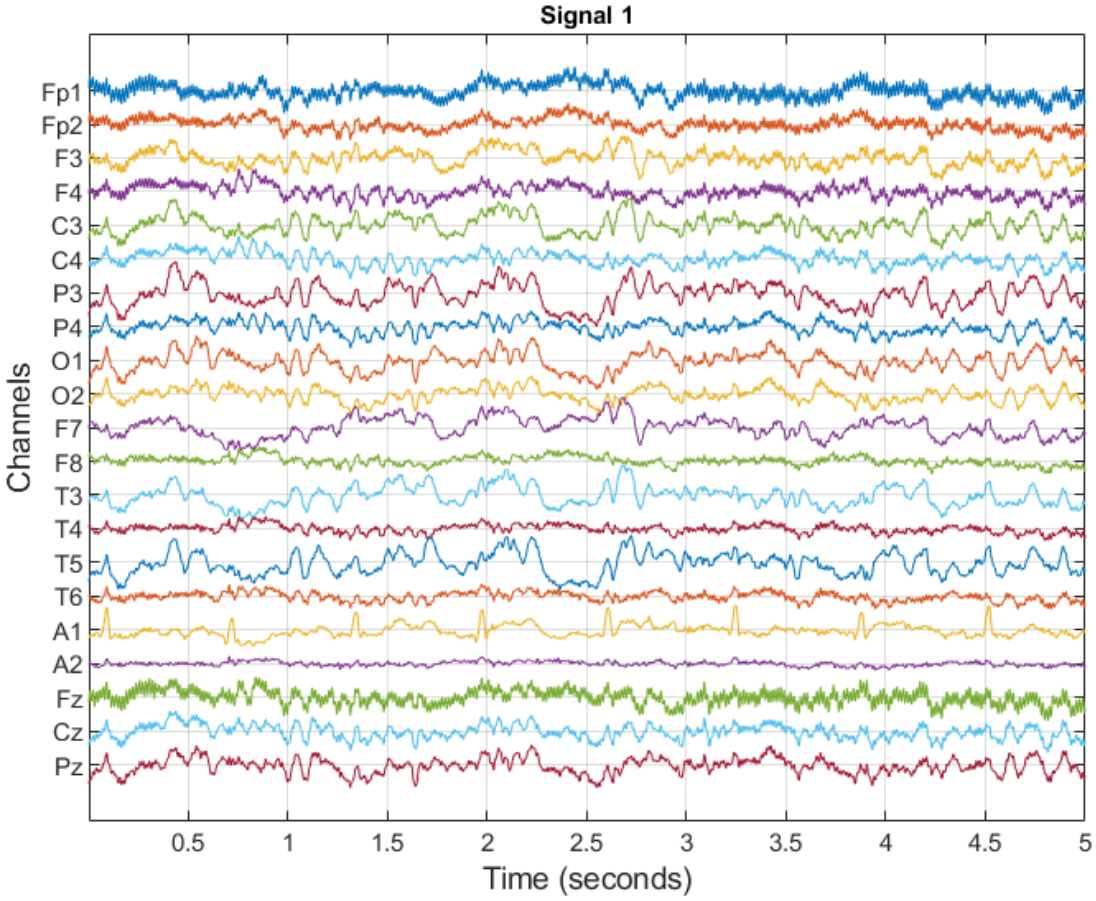

Figure 10: Plot of Signal 1

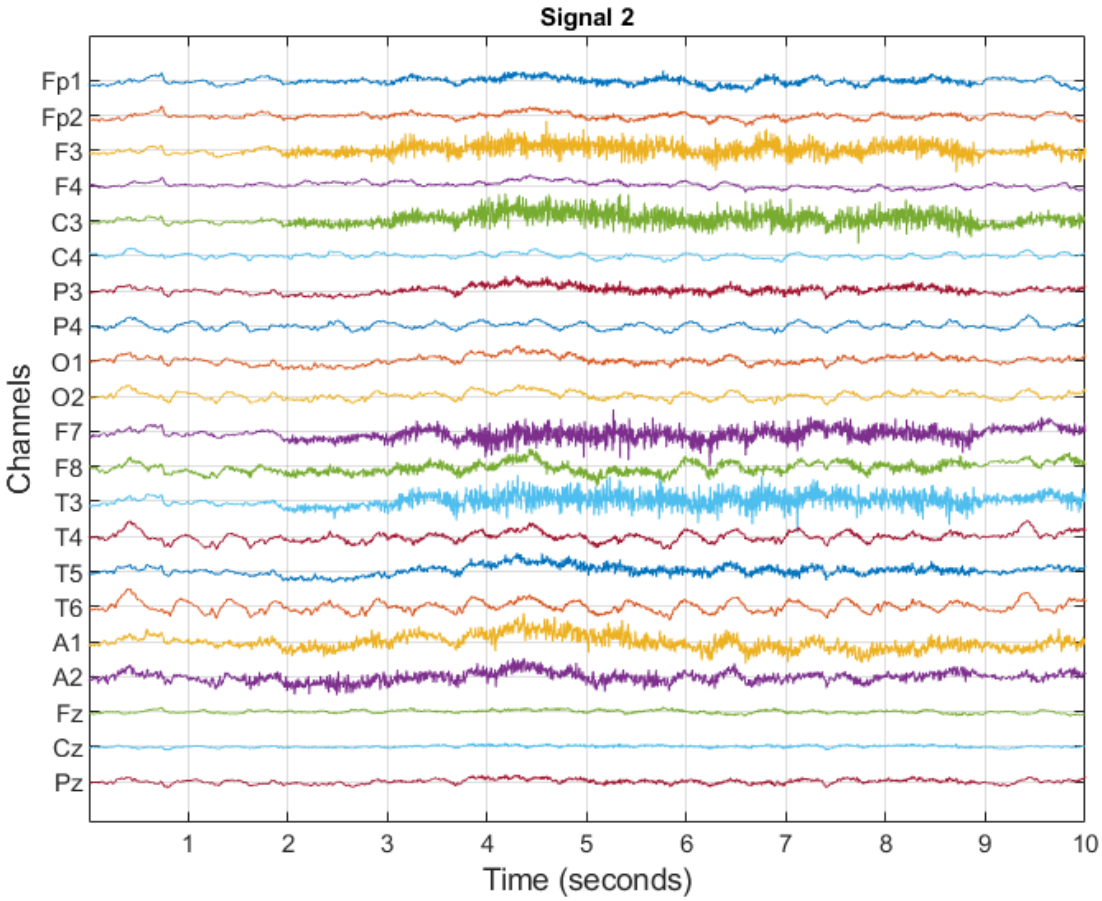

Figure 11: Plot of Signal 2

Question 2

Review the signal from the perspective of noise and artifacts. What type of artifacts and noise do you see? In your opinion, is this artifact removable using the ICA method?

Soloution

After examining both EEG signal displays, several types of artifacts and noise can be identified:

1. **Muscle artifacts:** Particularly evident in Image 2, channels F3, C3, F7, and T3 show high-frequency, irregular activity starting around 2-3 seconds and continuing for the rest of the recording. This is characteristic of muscle tension or movement artifacts.

2. **Eye blink artifacts:** In Image 1, there are occasional large amplitude deflections in the frontal channels (Fp1, Fp2) that are likely due to eye blinks.

3. **Electrode movement/pop:** Image 2 shows sudden, sharp changes in the baseline of some channels (e.g., T3, A1) which could indicate electrode movement or brief loss of contact.

4. **Power line interference:** There appears to be some high-frequency noise consistently present across most channels in both images, which could be 50/60 Hz power line interference.

5. **Slow drift:** Some channels in both images show gradual shifts in the baseline over time, which could be due to slow drift artifacts.

Regarding the use of Independent Component Analysis (ICA) for artifact removal:
ICA is generally effective for removing artifacts that have distinct spatial and temporal patterns that are independent of the underlying brain activity. Based on the artifacts observed:

- **Muscle artifacts:** ICA can often separate muscle activity, especially if it's spatially localized. However, the widespread muscle activity in Image 2 might be challenging to completely remove without affecting some brain signals.
- **Eye blink artifacts:** These are excellent candidates for ICA removal, as they have a characteristic spatial pattern and are typically independent of brain activity.
- **Electrode movement/pop:** ICA can potentially isolate these if they're infrequent and spatially specific.
- **Power line interference:** While ICA can sometimes isolate power line noise, other methods like notch filtering might be more appropriate.
- **Slow drifts:** ICA may be able to isolate these, but other methods like high-pass filtering might be more suitable.

Question 3

Apply the ICA algorithm on the signal and obtain the independent components and the mixing matrix. You can use the result of (Com2) from COM2R.m (an ICA algorithm).

```
Pest = size(signal,1);
[F,W,K] = COM2R(signal,Pest);
Z = W*signal;
```

Source Code 10: Question 2 - Part 3

Soloution

Z is Independent Components matrix and F is mixing matrix.

Question 4

Plot the signal in the time and frequency domains for each component, and in the case of undesirable results, make a decision. For frequency and spatial analyses, you can use the result of the pwelch.m function in order.
Use `plottopomap.m` for spatial and frequency analysis of the matrix of selected sources and provide comments regarding the spatial layout related to the mixing matrix.

```
% plot time domain of the components
offset = max(max(abs(Z)))/1;
feq = Fs;
```

```
    ElecName = 1:32;
    disp_eeg(Z,offset,feq,ElecName);
    xlim('tight')
    title('Components')
    ylabel('Components N');

    % fft
    figure('units','normalized','outerposition',[0 0 1 1])
    for i = 1:21
        subplot(7,3,i)
        windowL = gausswin(128);
        overlap = length(windowL)/2;
        L = length(Z);
        f = (0:(L/2-1))*Fs/L;
        [pxx, f] = pwelch(Z(i,:),windowL,overlap,f,Fs);
        plot(f,pxx,'LineWidth',1.5,'Color','#0072BD');
        xlim([0 70]);
        title("Component " + i)

        if(mod(i,3) == 1)
            subplot(7,3,i)
            ylabel('Magntiude');
        end
    end

    subplot(7,3,19); xlabel('Frequency(Hz)');
    subplot(7,3,20); xlabel('Frequency(Hz)');
    subplot(7,3,21); xlabel('Frequency(Hz)');

    % topography
    elocsX = Electrodes.Electrodes.X;
    elocsY = Electrodes.Electrodes.Y;
    elabels = Electrodes.Electrodes.labels;
    figure('units','normalized','outerposition',[0 0 1 1])
    nsub = 1;
    for i = 1:21
        subplot(4,6,nsub)
        plottopomap(elocsX,elocsY,elabels,F(:,i))
        nsub = nsub + 1;
        title("Component " + i)
    end
```

Source Code 11: Question 2 - Part 4

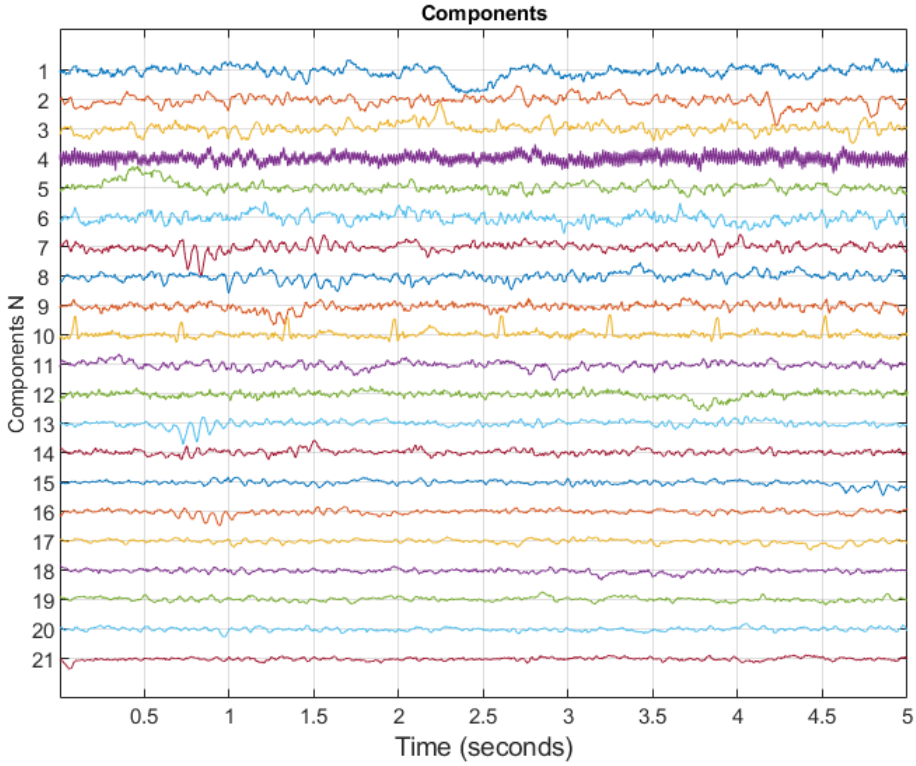

Figure 12: Time Plot of Signal 1 Components

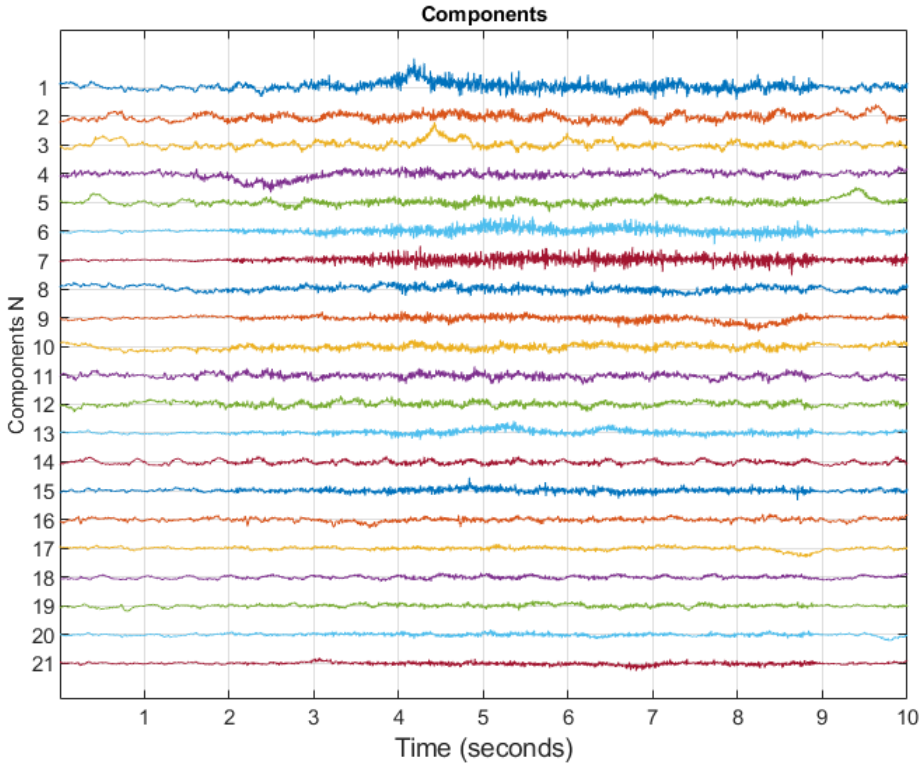

Figure 13: Time Plot of Signal 2 Components

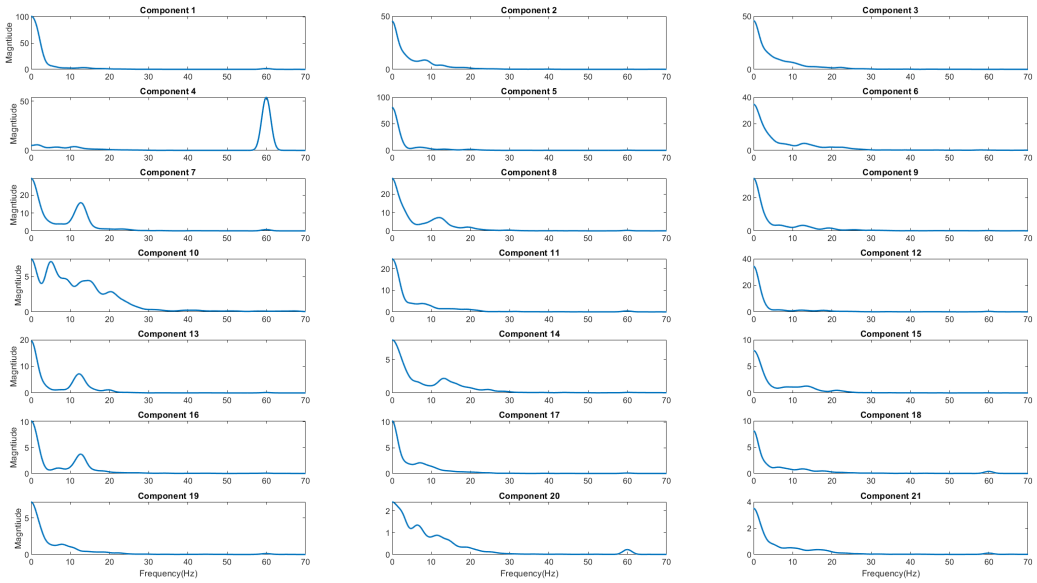

Figure 14: Frequency Plot of Signal 1 Components

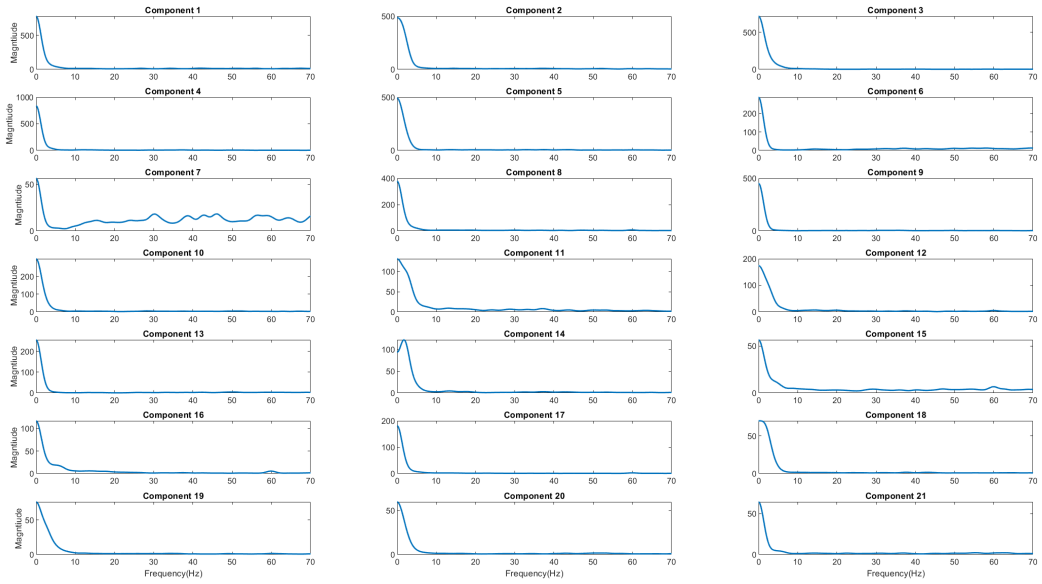

Figure 15: Frequency Plot of Signal 2 Components

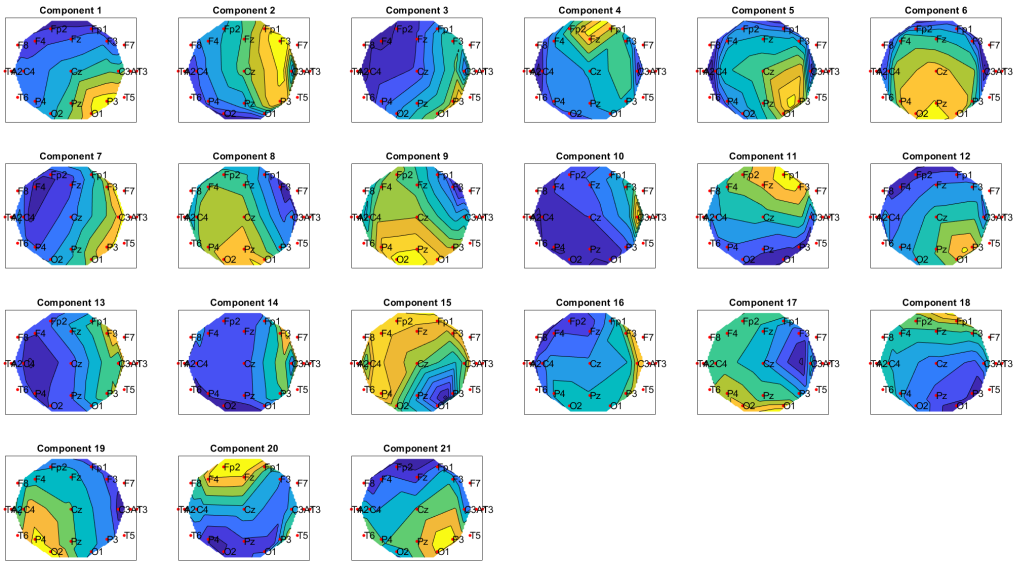

Figure 16: Topomap of Signal 1 Components

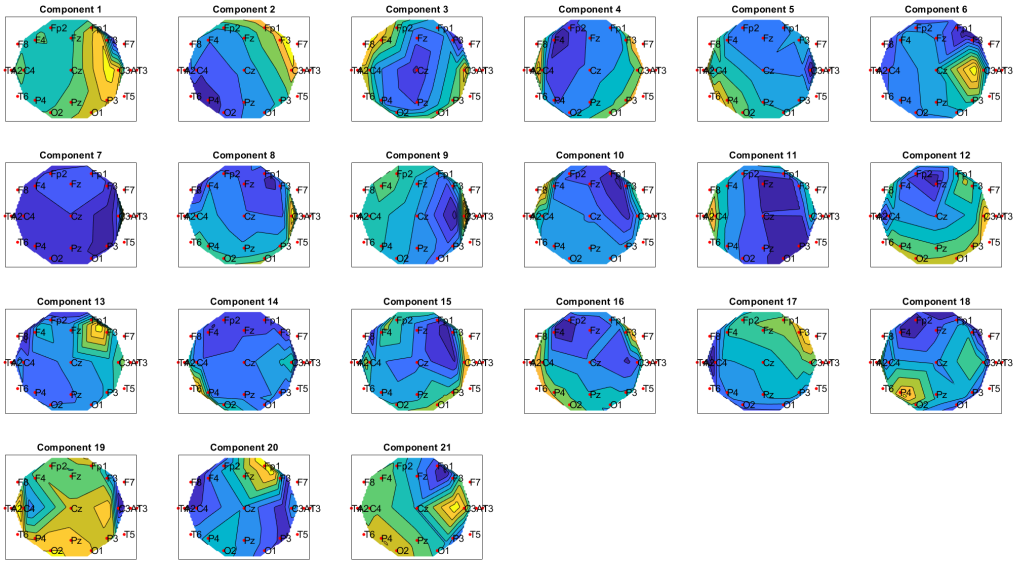

Figure 17: Topomap of Signal 2 Components

Soloution

In signal 1, components number 4 and 10 are not appropriate. We remove them. In signal 2, ICA could not find independent components well and we do not continue with TA's Permission.

Question 5

Identify all desired sources by saving them in SelSources, and retrieve the denoised signal as follows:

$$X_{\text{denoised}} = A(:, \text{SelSources}) \times S(\text{SelSources}, :)$$

```
rmv_comp = [4, 10];
Z(rmv_comp,:) = [];
F(:,rmv_comp) = [];

% reverse problem
X_Den = F*Z;

% plot
offset = max(max(abs(X_Den)))/1;
feq = Fs;
ElecName = Electrodes.Electrodes.labels;
disp_eeg(X_Den,offset,feq,ElecName);
title('Denoised')
```

Source Code 12: Question 2 - Part 5

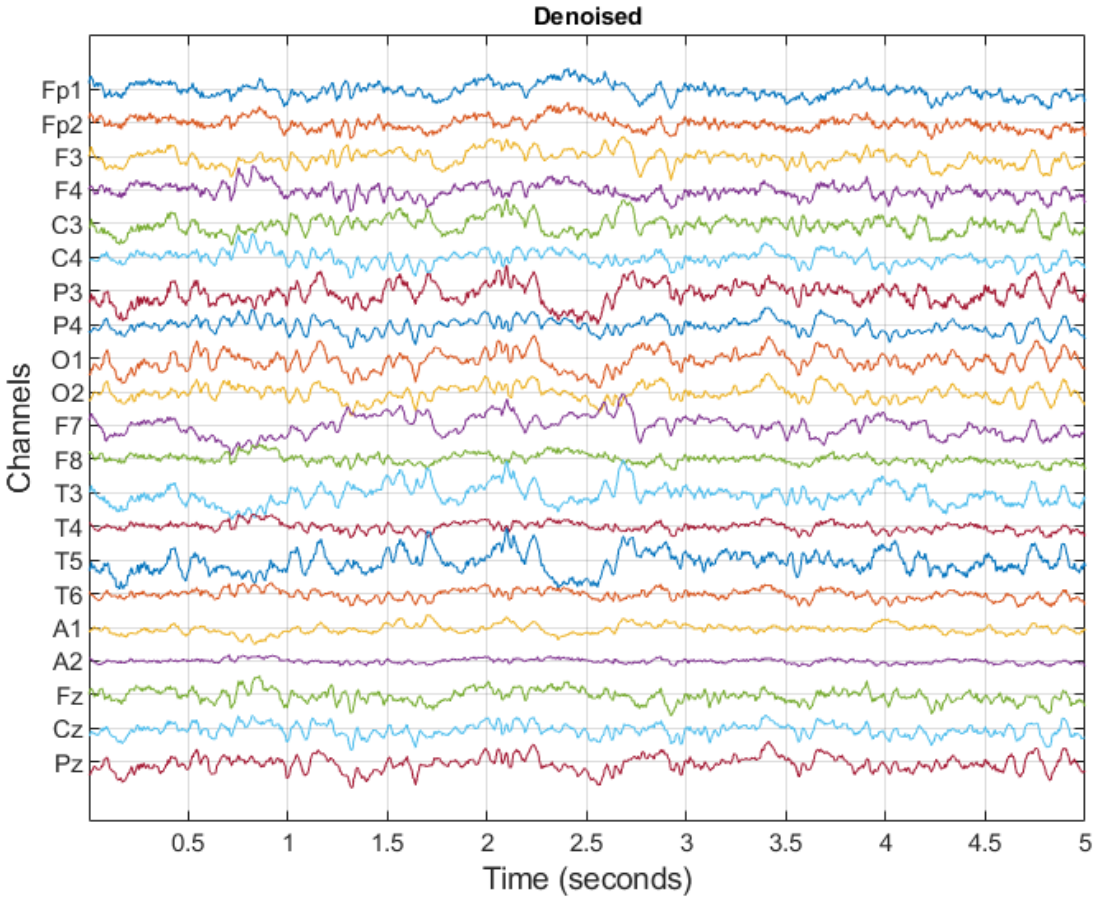

Figure 18: Signal 1 After removing artifact components

Question 6

Compare the denoised signal with the original signal. Have you chosen the sources correctly? Did you retrieve the noise-free signal in the first step, or do you need to reselect the sources more carefully?

```
% plot
offset = max(max(abs(X_Den)))/1;
disp_eeg(X_Den,offset,feq,ElecName);
title('Denoised')
offset = max(max(abs(signal)))/1;
disp_eeg(signal,offset,feq,ElecName);
title('Raw')
```

Source Code 13: Questio 2 - Part 6

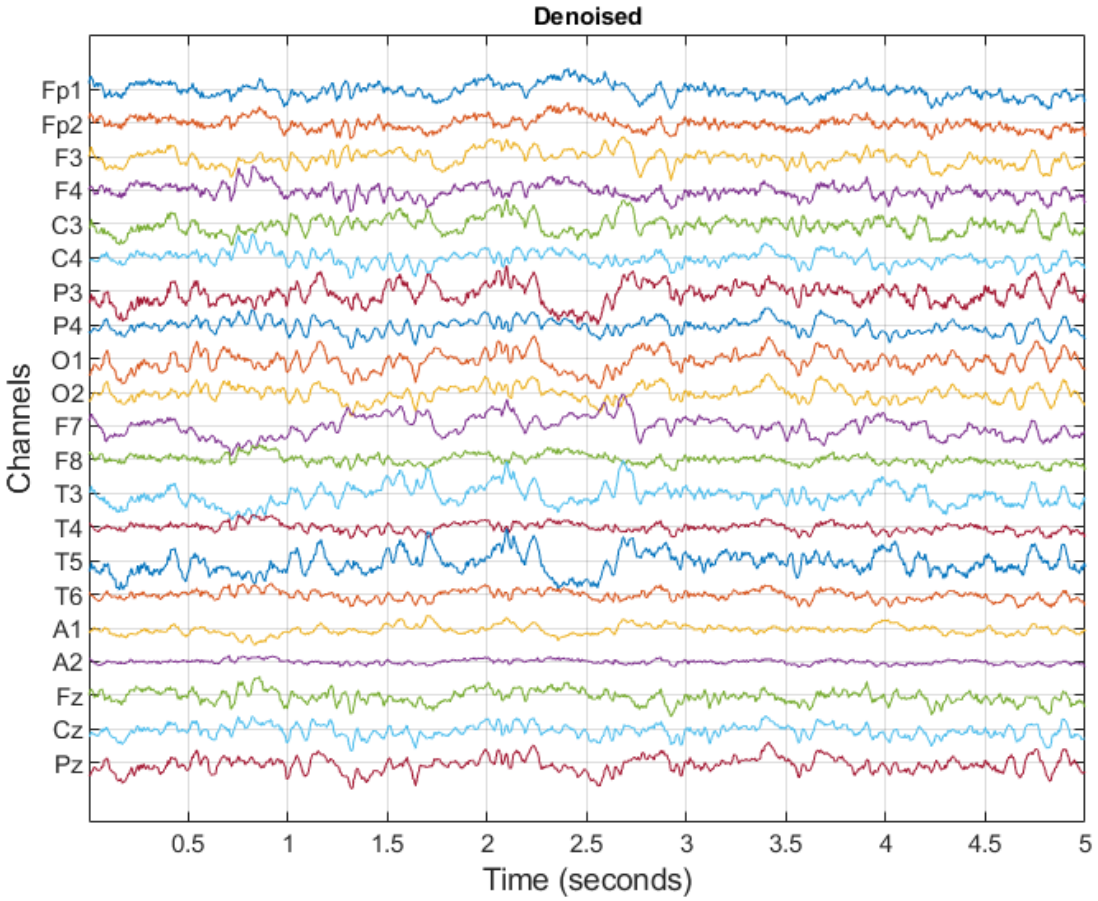

Figure 19: Signal 1 After removing artifact components

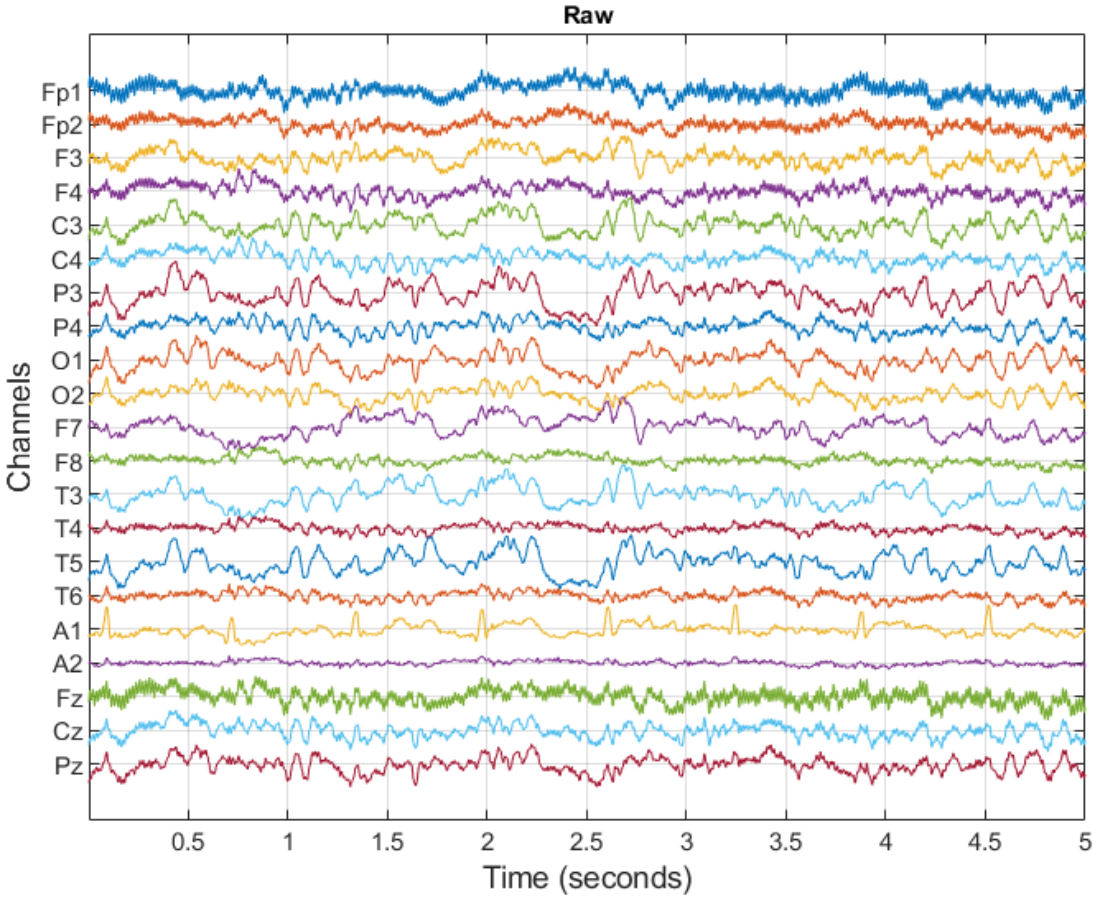

Figure 20: Signal 1 before removing artifact components

Soloution

I think we could remove noises well and we do not need to remove other components.