EEG artifact tool based on Multi-channel Wiener filtering: manual

January 2, 2020

1 Introduction

This document is a manual for the EEG artifact removal tool found at `https://github.com/exporl/mwf-artifact-removal`. Its purpose is the manual annotation of EEG artifacts caused by eye blinking, motion, etc. With these manual annotations, a multi-channel Wiener filter (MWF) can learn the statistical properties of these artifacts to extract and subsequently remove them from the signal. The application is started by opening the ***mwf-artifact-removal\gui*** folder in Matlab and running the ***main_gui.m*** script. This document will elaborate on the workflow of applying this algorithm using this tool and explain the workings of the various functionalities included.

2 Quick start on demo data

Firstly, we present a short demonstration on how the GUI can be used on the demo data that is loaded by default upon opening the GUI. A more detailed explanation of all the functionality the tool offers follows afterwards. We can already clearly see an eyeblink artifact in the first half second, especially in the frontal channels. That these artifacts are located there can also be seen on the topoplot below, showing the highest concentration of power in this region. To remove these eyeblinks, we will have to provide the MWF with fine markings of multiple similar artifacts, so we proceed as follows:

- Adjust the **Range** field to 25 seconds. We can now see about 5 of these artifacts, which should be enough.

- To see the artifacts in more detail, plot only the first 8 channels by filling the **Channels** field with 1-8.

- We now proceed to the marking process. The preloaded demo data also comes with an example annotation, which can be edited with the **Edit Markers** button. If you only want to see the effect of the MWF on the data, you can keep the default annotation and skip these next two steps.

- For the purpose of this example, we can start the marking from scratch with the **Restart Marking** button. A warning appears asking if we are certain we want to remove all current markers, click yes and proceed.
- Create a new marker for the first artifact by either right-clicking on the plot and selecting **New Marker**, or clicking on the **Add Marker** button. You can now drag a rectangle around the artifact, whose width, height and position can be adjusted at any point. Note that the height does not really matter for the annotation process, only its base should just surround the artifact. Repeat the process for the other artifacts. The algorithm will only consider data up to the last marking for the learning of the filter. All unmarked segments BEFORE the last marking are treated as 'clean'. Thus, be sure that all artifacts before the final marking are annotated to avoid leaking artifact samples in the 'clean' examples. Any erroneous marker can be deleted by right-clicking it and selecting **Delete**. When you are satisfied, click the **Confirm Marking** button.
- Now, the effect of the marking process is shown by giving artifact data a red shade and artifact-free data a green shade. This can be disabled again by clicking the **Show Marking** checkbox.
- We can now compute the Wiener filter and filter the data by clicking the **Filter Artifacts** button. This will use the default MWF settings, which can be seen when clicking the **Filter Options** button.
- The cleaned signal now appears in the bottom plot. To compare the original and cleaned data and see the effect on artifacts that weren't used for the filter computation, change the **Range** field to 30 seconds and click the **Overlay** checkbox. You can try to improve the performance by adding more annotations or improving the current ones with the **Edit Markers** button or playing with the MWF options.

3 Main GUI

The start screen consists of two plots on the left hand side and a set of buttons and boxes on the right hand side. The upper plot shows the input EEG data while the lower will show the output of the processing. In the right hand corner, a miniature topoplot can be found of the input data, with the energies being computed over the range of the top plot. Changing the axis limits of this top plot will automatically update the topoplot. The standard workflow will consist of setting the desired plot settings, editing the artifact markings and finally performing the artifact removal.

3.1 Plot Settings

The buttons in this section are:

- **Range** This input box allows the user to specify the range of the EEG plot in seconds.
- **Channels** This input box allows the user to control which channels are plotted and in what order. The required channels are separated by comma's. A range of channels can be indicated with the '-' symbol. Thus, to plot channels 1,3,12,13,14 and 18, one would enter 1,3,12-14,18. To plot all the channels, the user simply enters 'all'.
- **Time Slider** This slider allows the user to scroll through the EEG data in the time domain.
- **Scale Slider** This slider allows the user to change the scale of the plotted data. Using this slider effectively changes the distance between the axes around which each channel is centered. Thus, the outer left of this slider represents plotting all EEG's on top of each other, while the right means the data from each channel is perfectly separated from each other.
- **Channel Names** This allows the user to switch between showing the channel indices or their names in the plots. This option is only visible when the user has loaded a .mat file containing the labels of the channels.

Note that any change using the buttons will change both the upper and lower plot simultaneously. It is also possible to use the standard Matlab functionality to zoom in, zoom out, pan and reset. These are activated by clicking on the relevant plot. This will cause these symbols to appear on top of the figure, enabling and disabling them happens with a mouse-click. Note that these changes are **not** propagated to the other figure.

3.2 Marking

The buttons in this section are:

- **Edit Markers** This button starts the artifact marking process. The current artifact markers are indicated by rectangular regions that can be edited by dragging. This allows the user to change their width, height and position (though their height is irrelevant for the marking process). Markers can be removed by right-clicking them and selecting the 'delete' option. Starting the marking process also causes two new buttons to appear: the **Add Marker** and **Confirm Marking** button. **Add Marker** can be used to create a new marker, while **Confirm Marking** ends the marking process, saves the markers and restores the screen to its original view.
- **Restart Marking** This button acts in the same way as the previous one, but removes all the current markers.
- **Show Marking** This checkbox indicates whether the markers should be indicated in the upper plot. It is inactivated during the marking process

itself. Data in green shades represents 'clean' data and red shades represent artifacts. Data not shaded in any colour is not provided to the MWF in any way.

During the marking process, the plot settings can still be changed at will. However, to continue, it is very important to confirm the marking as the markers are not permanently stored before this action. The algorithm will only consider data up to the last marking for the learning of the filter. All unmarked segments BEFORE the last marking are treated as 'clean'. Thus, be sure that all artifacts before the final marking are annotated to avoid leaking artifact samples in the 'clean' examples.

3.3 Artifact functions

In this section we finally find functionality regarding the MWF artifact filtering. The last two buttons in this section are concerned with the generation of a topoplot of the data, and thus require a separate file containing the x-y locations of the channels. If this file is not present, these butttns are not visible.

- **Filter Artifacts** This performs the MWF algorithm and produces both the artifact signal and the cleaned signal (composed of the difference between the clean and artifact signal). The output is plotted in the bottom plot.

- **Filter Options** This opens a small menu where some options of the MWF can be adapted. These consist of:

  - **Delay** This setting determines the order of the temporal part of the MWF. Setting it larger will often produce better results, but at a larger computational cost
  - **Rank** The computation of the MWF involves the computation of a covariance matrix. The algorithm transforms this covariance matrix to a lowerdimensional space by discarding some of the eigenvalues in a number of possible ways. 'Full' changes nothing at all, 'Positive' only keeps positive eigenvalues, 'Percentage' keeps the X % largest eigenvalues and 'First' keeps the X largest eigenvalues. The X of the latter two methods can be filled in the box appearing when one of the two methods is selected.

  These changes are stored by clicking the **OK** button, reset to the way it was when entering the menu with the **Reset** button and restored to their default (recommended) settings with the **Default** button. After changing the options, the **Filter Artifacts** button should be pressed to recompute the artifacts.

- **Artifact/Clean** This dropdown menu allows the user to either select the artifact signal or the clean signal (the original signal with the artifacts removed)

- **Overlay** Enabling this checkbox overlays the computed signal with the original signal in the top plot.

- **Generate Topoplot** This opens the Topoplot GUI (see section 3).

- **Enable Topoplot** This enables/disables the miniature topoplot in the right hand corner of the screen. Disabling this will increase the speed of the rest of the application.

3.4 Menu

In the left-hand corner of the start screen, a drop-down menu can be found which can be used to load and save data. The format of these data files should be equal to the one used in the demo files. Possible files to be loaded includes:

- **Data** This consists of EEG input data. This should be stored in a .mat file with two fields. The first is a $N \times L$ matrix containing the EEG data, with N the amount of channels and L the length of the signal in samples. The second is the samplerate, expressed in Hz. An example (which is also the data loaded by default when starting the GUI) can be found in the **gui_demo_data.mat** file.

- **Markers** This allows the user to load previously stored markings. This .mat file consists of only one field, an $N \times 2$ matrix, with N being the amount of markers. Each row indicates the start and end time of an artifact marker. An example can be found in the **gui_demo_markers.mat** file, which is loaded by default when the GUI is opened.

- **Labels** This allows the user to add the channel labels of the data as extra information. This .mat file consists of a single field called labels, a string array containing a label for each channel. The default file for the demo is the **gui_channel_labels.mat** file.

- **Locations** Loading a locations file enables the use of the topoplot functionality. It contains a field called circumference, storing the circumference of the person's head in mm, an field called xi with the x-locations of the electrodes in mm and a field called yi with the y-locations of the electrodes in mm. The default file for the demo is **gui_64_Channel_locations.mat** file. This data originates from the 64-channel Emotiv-Epoc+ EEG headset.

Possible data to be saved includes the cleaned signal, the extracted signal and the current markers. The format in which they are saved is identical to the formats described above

4 Topoplot GUI

The Topoplot GUI allows the user to analyze the EEG data in a different way, by plotting the distribution of the EEG energy across the scalp. An important note is that the user should have a file included containing the x-y locations of the electrodes as described in the previous section. The first part of the GUI is once again concerned with the plot settings and includes:

- **Channel Names** This allows the user to switch between showing the channel indices or their names in the topoplot.
- **Original/Artifact/Clean** This drop-down menu allows the user to select the signal of which the topoplot is generated.
- **Frequency range** This allows the user to select only a certain part of the spectrum for which the energy will be computed. The lower and upper limit is separated by a '-'. For example, the user enters '1-4'. The user can also enter 'all' to indicate the entire frequency spectrum.
- **Time** This shows the user the time at which the topoplot is computed. To select a certain time him/herself the user can manually enter the starting time.
- **Range** The energy is averaged across a certain time span, which the user can adapt here.
- **Slider** This allows the user to slide the topoplot through time.

To store a representation of the topoplot across the entire dataset, the user can generate a GIF of the topoplot at some discrete points in time. This GIF uses the same settings as above with a few extra options

- **Filename** The name of the GIF to be stored. It should always end in '.gif'.
- **Framerate** This indicates the amount of frames shown per second.
- **Frame overlap** Each frame consists of the topoplot taken across a certain time window. This field indicates the amount of overlap between each successive window. A higher overlap results in a smoother video, but with more frames.
- **Generate GIF** Once all these settings are set, the user presses this button to generate the GIF. A progress bar will appear, also allowing the user to abort the GIF generation at any point. The GIF will then contain only the frames computed up until that point.