EEG-EMG during gait: acquisition and analysis

1st Tommaso Urbani
dept. of Information Engineering
University of Padua
Padua, Italy

2nd Enrico Fongaro
dept. of Information Engineering
University of Padua
Padua, Italy

***Abstract*—This project aims to acquire and analyse EEG and EMG data during gait, in order to study the brain and the muscular activities during its cycle. The acquisition is performed using the ANTNeuro (EEG system) and the COMETA (EMG system) devices on a healthy patient. The EEG-EMG signals are therefore preprocessed in order obtain a better representation by frequency filtering and removing artifacts.**

***Index Terms*—EEG, EMG, Cometa, ANTNeuro, gait**

INTRODUCTION

This work shows the acquisition and the preprocessing steps of the EEG and EMG signals during gait. The first chapter describes the setup adopted in the lab activity, the ANTNeuro (EEG system) and the COMETA (EMG system) devices with insertion of a triggerbox in order to guarantee synchronization between the two systems. The second chapter provides useful information about the software acquisition, enabling the user to manage the eego and EMGandMotionTools software, respectively for the EEG and the EMG data, and trigger.py algorithm. The third chapter allows to understand the working acquisition steps during the gait, providing a guideline of all the actions that have been performed. The fourth and the fifth chapters show synthetically the pipeline adopted for the preprocessing of the EEG and the visualization of EMG dataset, carried out respectively by using EEGLAB and an other software named MLSviewer. The sixth chapter shows the relation between the EEG and EMG signals during the gait analysis, accordingly to specific onset and offset specified by the triggerbox. Finally, the seventh chapter described the limitations of the setup and some possible solutions in order to avoid many difficulties about connecting devices and enhance a simpler acquisition.

I. SETUP

The hardware setup is composed by: the Cometa amplifier, the EMG sensors, the triggerbox, the AntNeuro, the EEG cap and all the cables. Here you can find the steps to build the setup for each system:

Cometa

- Screw the three antennas into their connectors
- Connect the T-C cable from the Cometa amplifier to the Triggebox in order to start the acquisition synchronized with the AntNeuro.
- Connect the C-Tablet cable from the Cometa amplifier to the tablet to visualize the live data from each EMG sensor.
- The Cometa amplifier need to be switched on, there is a ON/OFF button on the back.
- The EMG sensors communicate with the Cometa amplifier through a wireless connection.

AntNeuro

- The AntNeuro amplifier needs to be fully charged before the acquisition, to charge it there is a power cable.
- Connect the T-A cable from the AntNeuro to the Triggerbox in order to start the acquisition synchronized with the Cometa
- Connect the A-Tablet cable from the AntNeuro to the tablet to setup the software settings from the acquisition (next chapter).
- The AntNeuro electrode cap needs to be connected to the amplifier with A-cap1 and A-cap2 cables (the order is important).

Triggerbox

- The Triggerbox has 2 power cables that need to be always connected to the power supply.
- Connect the T-PC cable from the Triggerbox to the PC in order to start the Triggerbox software.
- The T-A cable should be connected from the Triggerbox to the Ant Neuro.
- The T-C cable should be connected from the Triggerbox to the Cometa amplifier.

Tablet

- The tablet needs to be fully charged before the acquisition, to do so there is a power cable.
- Connect a hub USB to the PC with at least 2 USB ports.
- The C-Tablet cable should be connected from the PC to the Cometa amplifier.
- The A-Tablet cable should be connected from the PC to the AntNeuro.

PC

- The PC needs to be connected with the triggerbox through the T-PC cable in order to start the trigger by running the program *triggerbox.py*.

The configuration should follow the setup in Figure 1.

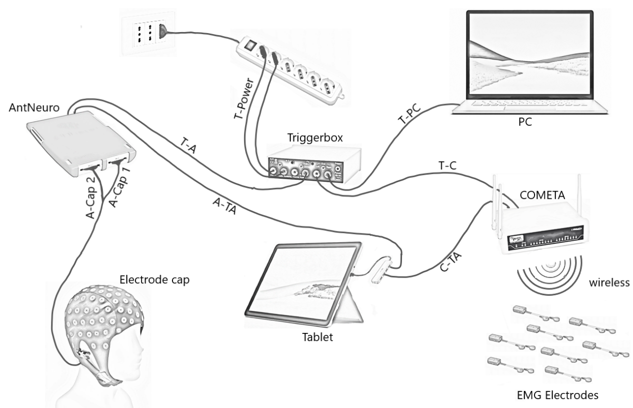

Fig. 1. Setup map for the EEG-EMG acquisition

II. SOFTWARE

eego and EMGandMotionTools are respectively the two software corresponding to the EEG (AntNeuro) and EMG (Cometa) signals, used to record, process and visualize the data. Triggerbox.py is a python algorithm which provides the synchronization between the AntNeuro and Cometa devices.

eego

eego software is a powerful and intuitive tool for recording and online processing of EEG data. Data from multiple sources (up to 64 sensors) of the electrode cap are automatically synchronized and stored together in a single database. A list of all the steps performed during the data acquisition is presented with the purpose of facilitating the user:

- start the eego software and make sure that the eego amplifier is connected, checking the battery state, in the lower right part of the interface program.
- *Edit Montage* allows to start a workflow to customize the montages
  - create a new montage by choosing the sensors of the electrode cap of interest: depending on the purpose, you can choose a subset of the total 64 electrodes
  - keep in mind that the referential channel must be the same for all electrodes, e.g. you can set the mastoid (M) or the Cpz channels.
  - other possible features related to filtering approaches can be used such as cutoff frequencies and notch filtering or other data manipulations for improved real-time viewing, while always storing unaltered raw data for offline analysis
- *Edit Amplifier* shows and allows modifying amplifier setup which can easily and reliably be configured to suit the requirements of a specific experiment or diagnostic.
  - Create a new amplifier setup by modifying the relevant parameters contained in its configuration, such as type of the amplifier channel, electrode names, signal range per channel, name of reference electrode
  - or Select an amplifier from the previous sessions.
- *Acquire* allows you to record data from your eego amplifier and visualize the data in real-time on the screen:
  - Select an existing subject or create a new subject
  - Select the amplifier setup: choose the amplifier name during the *Edit Amplifier* phase
  - Select the proper montage for data display: choose the montage setup during the *Edit Montage* phase
  - Select the sampling frequency: 2048 Hz is a good choice of this parameter
  - Open a new workflow page dialog by pressing Next
  - Take the syringe and gently fill the electrodes of the cap with electrolyte gel. The purpose of the gel is to create an electrical connection between the scalp and the electrode, reducing the electrical resistance (or impedance) between them. The button ”Impedance Check” displays the electrode impedance: values of 50 kOhm or below are sufficient to obtain high-quality EEG signals (green electrodes in Figure 2).

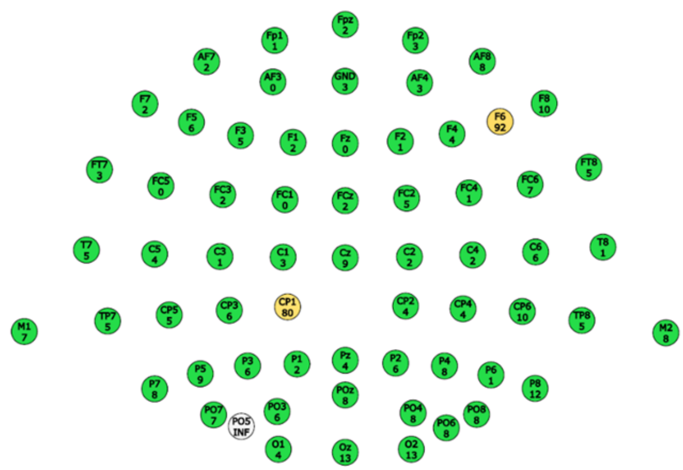

Fig. 2. Screen capture of the eegosoftware in EEG impedance check mode.

  - Record data by pressing the “Start Recording” button and ”Stop recording” (Figure 3).

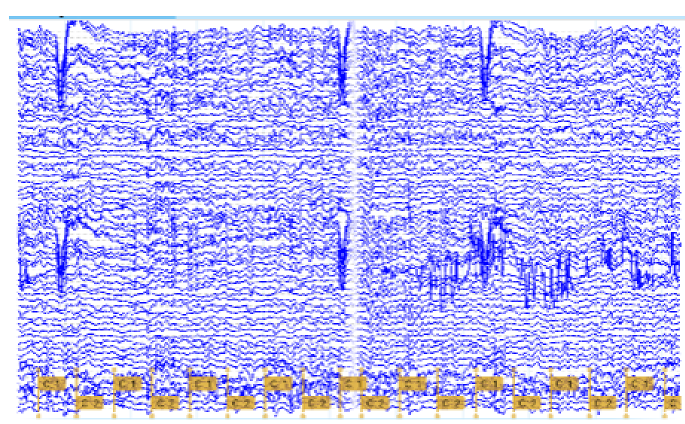

Fig. 3. The EEG View tab of the Acquire workflow

– Finally, complete the data storage to the database by pushing the “Finalize” button in the workflow page.

- *Review and Export* provides a quick and easy way to view data after a recording session, and enables exporting of data for analysis.
- Selecting a session to inspect all data you have recorded so far and directly export them.

EMGandMotionTools

EMGandMotionTools is a software developed by Cometa for EMG data recording and processing, represented as muscular activity acquired by multiple electrodes (up to 8 sensors) displaced on the subject’s body. EMG are automatically synchronized and stored together in a single database. The most important steps performed during the acquisition is presented below:

- by pushing the USB icon, the workflow page shows the EMG time series of the electrodes positioned on the body of the person. In the right part of the page, you can see the output values of the accelerometers contained in the electrodes.
- the play button starts the recording of the EMG signal for each electrode
- by pressing the settings icon, you can set or modify parameters such as:
  - the number of electrodes to acquire their signal and other possible chatacteristics of the sensors
  - set ”trigger in” instead of ”trigger out” in order to obtain the sinchronization of the EMG and EEG signals
  - the location where you want to save the EMG dataset
- by pressing the folder icon, you can save the dataset in .c3d format.

triggerbox.py

triggerbox.py is a python algorithm which enables the synchronization between the AntNeuro and Cometa devices, in particular it can be used to track the synchronous activity between the EEG and EMG dataset.

- keep in mind that this algorithm must be executed in the laptop (which is always commented with the triggerbox) and not using the tablet, in order to avoid the disconnection of the triggerbox when performing gait cycle
- start a cmd terminal in the laptop and run triggerbox.py algorithm
- you can choose any synchronization mode you want, from 1 up to 30: this number allows you to track a specific trigger marker if you need to consider multiple synchronizations
- when pressing your number, you can see, simultaneously, the appearance of the trigger marker with the corresponding number on the eego interface. You can not see in live the marker in the EMG time series but, after saving the dataset, the EMG signal starts from the trigger marker.
- by writing ”STOP” on the terminal, you can create the final trigger marker and the eego and EMGandMotion-Tools stop the acquisition and the recording of the two signals. The two datasets, containing the trigger markers, are saved.
- you can also check the number of triggers you have used during the acquisition by pressing the Finalize button on the eego software.

III. Acquisition

EEG

Before starting the acquisition you need to choose which montage to use, the standard acquisition setup is the International 10-20 EEG system, which is a recognized method to place the electrodes. The goal of this system is to place the first electrode (Cz) in order to obtain all the other electrodes spots for the best acquisition. To find the exact spot you need to take two measurements:

- the measurement from the *nasion*, which is the point between the eyes and above the bridge of the noise, and the *inion*, which is the crest point of the back of the skull.
- the measurement from left pre-auricular to right pre-auricular which is the point in front of each ear (in Figure 4 you can see the exact spot).

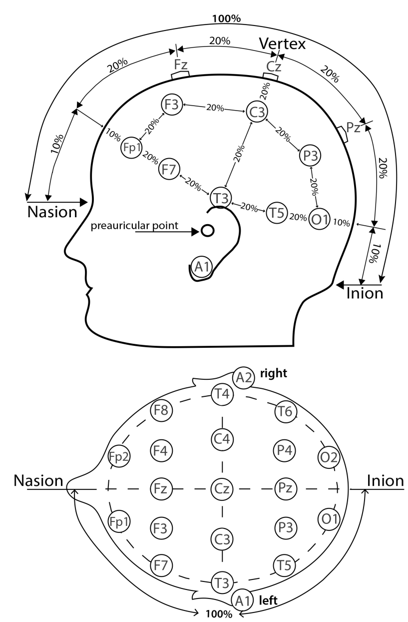

Fig. 4. International 10-20 system of electrode placement

The point where Cz will be placed is in the intersection with the middle of each measurement. Since you are using a cap with fixed distances between electrodes you need to place Cz in the spot you just found and then put on the cap, then be sure that it is aligned symmetrically with the eyes.
In Figure 4 we can see all the electrodes with their code, where F stands for frontal, P for Parietal, T for temporal, O for occipital, the numbers are odd for the left side and even for the right side and the electrodes with a Z are in

the midline. After the correct placement, you need to insert the electrolyte gel in each electrode of your montage with a syringe in order to create an electrical connection between the electrode and the skin. The first electrode you must fill is the GRN that represents the ground. Then you can proceed to fill each electrode with gel being careful to not overflow the electrode, this may cause a bridge between two different electrodes damaging the acquisition. You can check the quality of the impedance using the *eego* software as suggested before.

EMG

The EMG placement varies according to which task you want to study. In our case, since it is a gait analysis, we are interested in the lower part of the body. In Figure 8 you can see various placement points for EMG sensors.

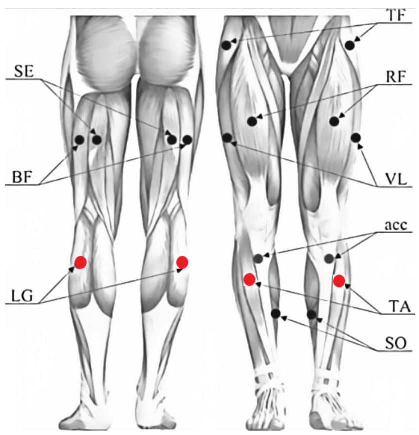

Fig. 5. The placement of 16 surface EMG sensors and two accelerometers over lower-extremity. For this project we focused just on TA and LG

We decided to use right and left anterior tibialis muscle and right and left lateral gastrocnemius muscle for gait EMG acquisition. First thing to do is to find the muscles, after that you can make a mark with a pen on the right spot. When you found the spot, you can clean it with some alcohol to obtain a less noisy signal. It would be better if the subject had no hair in that area. Then you need to clip the electrode in the EMG sensor and stick the electrode in the correct spot. In order to finalize the EMG placement, you need to secure with some tape the EMG sensor to the skin of the subject.

Triggerbox

Before the acquisition starts you need to check if the setup of the cables is right (Chapter II). Then you can run the program *triggerbox.py* in the PC, the program will detect the trigger box and will ask you to insert a command.

Starting the acquisition

Start the recording in the *eego* and *EMGandMotionTools* software by clicking the "play" button. When the trigger will be sent, the *eego* will leave a mark in that time and the *EMGandMotionTools* will start the real recording. To give the trigger impulse, you need to write the number 1 on the *triggerbox.py* program, after this you can disconnect the Cometa amplifier from the tablet and the AntNeuro from the triggerbox. At this point there should be left only the connection from the tablet to the AntNeuro, then the tablet and the AntNeuro are placed inside the backpack and the subject is free to walk. In our acquisition we asked the subject to walk for few meters, stop, make a 180 degrees turn, and walk to the starting point. It's suggested to wait a moment before starting each task to have a clear separation between the tasks and it's also suggest to take a video recording of the experiment to have a time reference to compare with the data. In Figure 6 you can see how the subject should be equipped during the experiment.

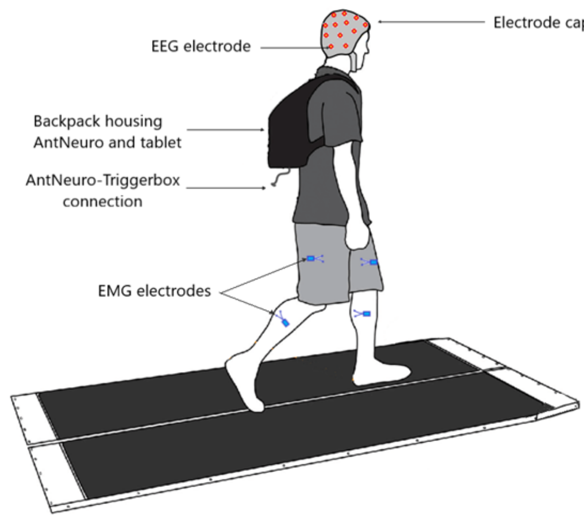

Fig. 6. Decription of the acquisition during the gait cycle

Ending the acquisition

When the subject completed all the tasks the experiment is completed, the subject can remove the backpack paying attention that the connection between the cap and the AntNeuro is still intact. The AntNeuro needs to be connected to the triggerbox and the Cometa amplifier to the tablet. When it's all connected you can write "STOP" on the *triggerbox.py* program and both acquisitions will stop.

```
Command: 1
         ---- WORKING - MODE:1 ----

Command: STOP
Previous acquisition has stopped
```

Fig. 7. *triggerbox.py* output

IV. EEG PREPROCESSING

EEG data out of the recording device is a continuous unprocess signal, represented by the difference of potentials between the electrodes and the reference according to the time. In order to extract meaningful measures from it (such as brain oscillations or brain source activations), we apply a series of preprocessing transformation to the data. The purpose is to clean the raw data aiming at extracting only the meaningful signal and reduce the amount of noise/artifacts affecting the output voltage. The figure 8 shows the final result of the preprocessing steps over all the electrodes/channels of the cap, where a clear description of the EEG signal is represented by the Cp1 channel.

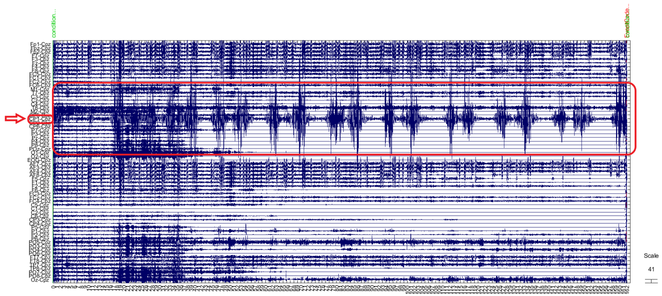

Fig. 8. EEG signal during gait acquisition after the application of preprocessing steps during gait

The tool used for achieving this objective is EEGLAB, a MATLAB toolbox for processing data from electroencephalography (EEG) and other electrophysiological signals. The general EEGLAB preprocessing pipeline described by Swartz Center for Computational Neuroscience and Makoto is the following:

a) Import data into EEGLAB: : after collecting the data, check the EEG file formats and importing the datasets using Neuroscanio (.eeg and .cnt files) and BIOSIG toolbox (.edf file). If event import was successful, you will see all the characteristics of the EEG dataset, among which `channels per frame` (n. of electrodes), `frames per epoch` (total n. of samples), `events` (trigger markers), `epoch end` (sensing time) and `sampling frequency`.

b) Re-reference data and resample data (if necessary/desired): : according to Makoto's work, the average referencing is, in theory, an approximation of scalp potentials that is independent of reference location of electrodes but in practice, depends on distribution of electrodes. For this reason it is not recommended when we use a subset of electrodes. Downsampling is useful to compress the data size by reducing the space and the time. You may downsample the data to a lower frequency if the original sampling frequency is higher than that.

c) Remove DC offset and baseline drift: : DC offsets introduce large filter artifact at signal boundaries, so it is better to remove them prior to filter the signal. Baseline drift represents a slow change in the baseline position, mainly caused by temperature effects on the eletrodes.

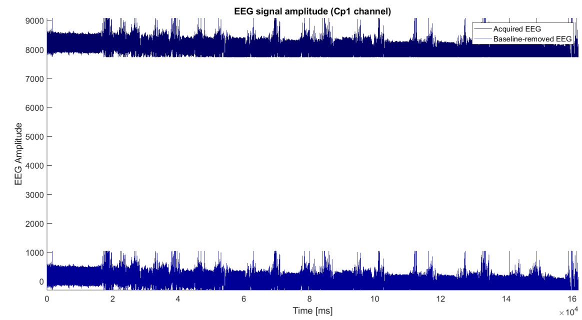

Fig. 9. Acquired EEG signal for the Cp1 channel and after its baseline remove

d) High-Pass Filter the data at 1 Hz: the reasons for the 1-Hz pass band edge is motivated by the fact that the most useful EEG information is in 3-45 Hz and low-frequency EEG (lower than 1 Hz) could be contaminated by sweating.

e) (Optional)Low-Pass filter and CleanLine: Low-pass filter is optional. If CleanLine, which uses an approach for line noise removal does not work well, you may want to simply apply a low-pass filter to cut out above 50/60 Hz. CleanLine works by setting the line noise frequencies to remove typically in the range [60, 120] Hz and uses sliding window to estimate sine wave amplitude to subtract.

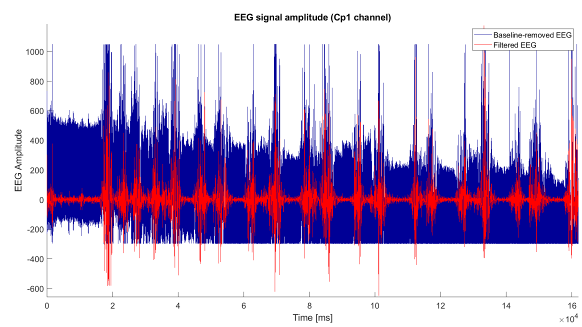

Fig. 10. Baseline-removed EEG signal for the Cp1 channel before and after filtering

f) Remove and interpolate EEG artifacts: The amplitude of artifacts (such as eye movements) is often larger than the amplitude of brain data which potentially decrease signal/noise ratio, bias data analysis and potential results. In this step, we are looking for possible bad channels in order to remove them. Removing bad channels can be performed manually by visual inspection or by using `clean_rawdata()` autodetection funtion. Note that this function saves nice results with which you can identify which channels and datapoints are rejected in the original length. For many reasons related to the average referencing, the rejected electrodes can be also interpolated, using spline interpolation, in order to avoid these possible issues.

g) Extract and reject epochs: reject epochs for cleaning the EEG signal. Aim for around 5-10% rejection rate. A simple strategy is to apply amplitude threshold of -500 to 500 μV (don't capture eye blinks), reject the selected epochs, and apply improbability test and reject the selected again. `clean_rawdata()` implements the Artifact Subspace Reconstruction (ASR) algorithm which finds clean portions of data and calculates PCA-extracted components' standard deviation, rejecting data regions if they exceed 20 times (by default) the standard deviation. Since we are not interested in rejecting bad channels (because we are just focusing on Cp1) and bad epochs (since epochs are relevant in gait analysis), for completeness we show the result of ASR applied to the electrodes signals, representing a standing person with open eyes (Fig. 11).

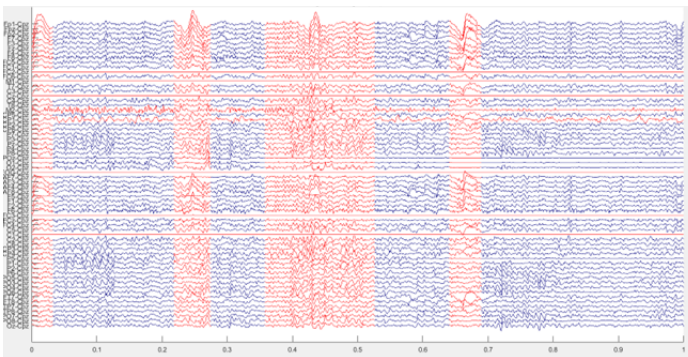

Fig. 11. Part of the EEG signals of a standing person with open eyes after applying ASR: red parts will be discarded.

h) Independent Component Analysis for artifact removal: Independent Component Analysis (ICA) may be used to remove/subtract artifacts embedded in the data (muscle, eye blinks, or eye movements) without removing the affected data portions. It is not implemented in this project but it is an important step within the EEG preprocessing pipeline when one is interested to find brain sources.

V. EMG SIGNAL

The EMG signals recorded by the *EMGandMotionTools* software is saved in the `.c3d` format. This is a public domain file format that has been used in Biomechanics, Animation and Gait Analysis laboratories to record synchronized 3D data. The format stores 3D coordinate and numeric data from any measurement trial, with all the various parameters that describe the data, in a single file. Since this format is more prone to the 3D movements than the EMG file, we found some difficulties to extract only the EMG signal, especially because there isn't an easy way to convert the data in a more common format like `.mat`. Since the goal of our project was only to visualize the EMG raw data of each sensor, we decided to search for a software that was able to do so, our choice was *MLSviewer*: a free C3D file viewer that displays the analog and 3D data present in any C3D formatted file, together with all of the parameters that are created and stored in the C3D files by C3D compatible applications. This software is very easy to use but is not able to modify the signal since it's only used to display the signal, also it's possible to view the movement of each sensor in all the three axes. In our acquisition we used 4 sensors: one in the right anterior tibial muscle (RTA), one on the left (LTA), one on the right lateral gastrocnemius muscle (RGL) and one on the left (LGL). The tibial muscles signals should have a peak when the subject is raising the toes, and the gastrocnemius musles signal when the subject stands up on tiptoes. Since we want to analyse the gait we should have a combination of this two movements when the subject will move the ankle. After the acquisition we obtained the signals that you can see in the next Figures:

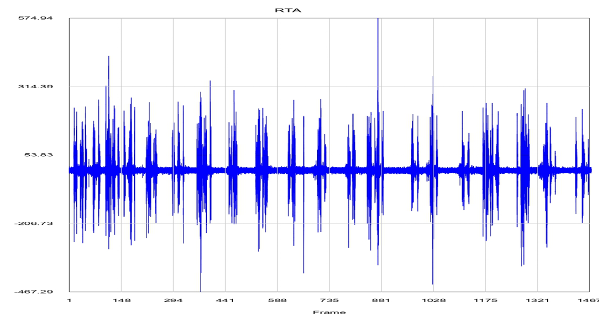

Fig. 12. RTA - Right Tibial Anterior EMG signal during gait

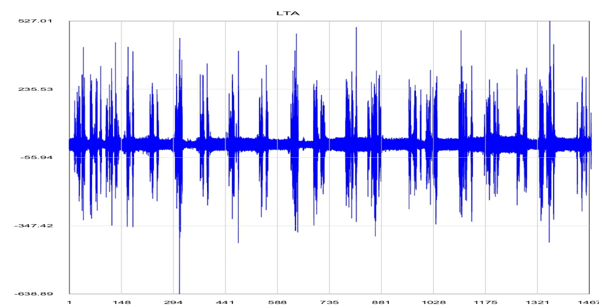

Fig. 13. LTA - Left Tibial Anterior EMG signal during gait

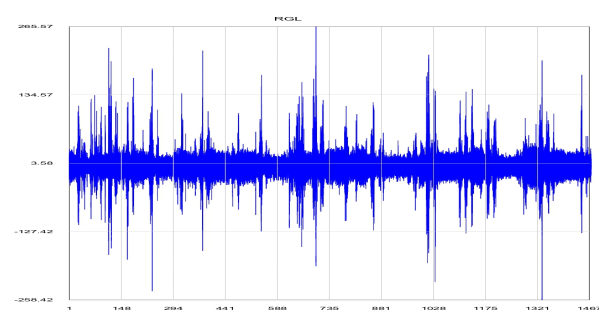

Fig. 14. RGL - Right Gastrocnemius Lateral EMG signal during gait

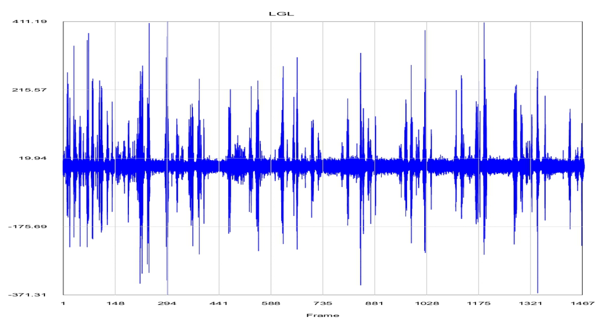

Fig. 15. RGL - Left Gastrocnemius Lateral EMG signal during gait

The signals shows us what we expected, for the Tibial Anterior muscles we have a peak for each step the subject took, since the starting of each step requires the raise of the toes. The Gastrocnemius muscles signals are not clear as the previous one, this is due to the usage of that muscle during a normal gait. In order to highlight better the signal, the subject should have changed the way of walking, using more the calves.

VI. Data Alignment

One of the main problem concerning data alignment between EEG and EMG signals is the lack of the initial trigger marker in the visualization of the EEG signal. We do not know the reason why, even if we started the trigger, its marker cannot be visualized. This problem leads to the impossibility to align the two signals in order to see the relation between them during their time course. However, the Figure 16 shows a course alignment based on the ideas that the trigger marker is supposed to be within the temporal interval [17, 19] s. This is motivated by three reasons:

- the fact that the trigger was started before the first EEG peak, meaning the first time the subject moved during gait acquisition
- the EMG dataset starts from the time the trigger has been run and ends when the trigger has been stopped. This allows to consider the idea that every impulse in the EMG signal is related to the corresponding impulse in the EEG signal

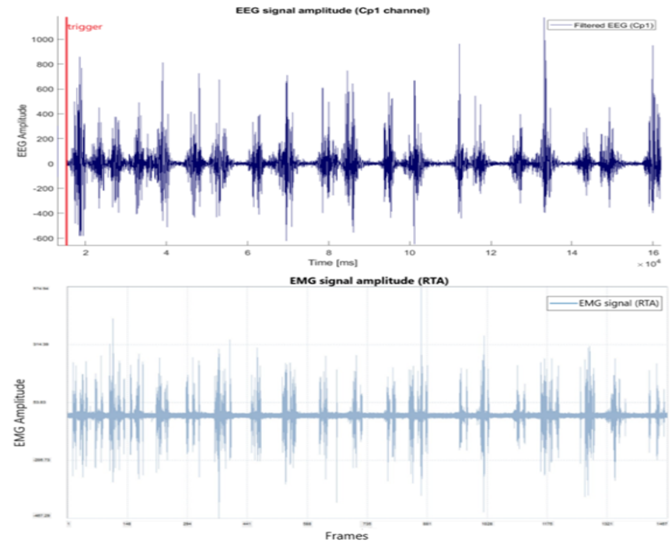

Fig. 16. Coarse alignment of EEG and EMG signals

Even though the alignment is very raw in term of used procedures, the final result shows that there exists correspondence in the two datasets since every impulse of the EEG matches with the corresponding impulse of the EMG. By the way, the lack of the synchronization marker between the two signals makes us unable to extract any other information between the two signals.

VII. Limitations and possible solutions

This setup allows to provide reliable physiological data in real-time during the acquisition but presents some limitations when performing gait analysis. All the recorded data are stored directly on the tablet which is carried in the backpack together with the amplifier. This aspect has either a great advantage or a disadvantage. The advantage regards the prevention of data loss. As the amplifier and the tablet are connected directly with each other, the data transfer process does not depend on the quality of the WiFi or the Bluetooth connection and therefore the data are always safely stored on the tablet. The disadvantage is that you always need to be connected to the tablet while you do the gait acquisition. There are also other different problems with the acquisition, such as some software issues like saving files through EMGandMotionTools and the fact we can't see the live data during the gait acquisition. Another big issue is the difficult of the actual acquisition, the backpack is not comfortable and problematic to use for different factors. Here we listed some suggestions to improve the acquisition method:

- customize the backpack to have an easier way to connect the cables to the tablet and the AntNeuro.
- in order to visualize the live data during the gait acquisition you can connect the tablet and a smart monitor to the same WiFi and screen mirroring the session. Another method is to create a Zoom call on the tablet and share screen the live data, or using AnyDesk software you can control the tablet with another PC.

Fig. 17. ANTNeuro backpack

- since the tablet is very fragile and it has only one USB port, we suggest to use a mini PC with battery. This solution increases the portability of the system, prevents accidental touches of the screen of the tablet and makes it easier to connect the cables.
- a simple and cheap solution to the accidental touches in the tablet is to apply a cover to the tablet.
- instead of using paper tape to secure the EMG sensor to the subject, it is easier to use double-sided tape.

New setup

Since we later found out that the setup presented does not work as we expected because the Cometa amplifier needs to be connected during all the acquisition to the device where the *EMGandMotionTools* software is running, we thought that a possible solution is to download the mentioned software in the same PC where the *triggerbox.py* program is running and to connect the Cometa amplifier to that PC. This solution will eliminate the need to run the *EMGandMotionTools* software in the tablet and especially the need to connect the Cometa amplifier to the tablet. Due to the short time available we cannot test this new setup but we are confident that this will improve drastically the portability and the robustness of the acquisition method.

VIII. Conclusion

This project gave us the opportunity to study and test a real EEG-EMG joint acquisition, it was clear from the beginning that some of the difficulties of this project were the practical ones, learn new software and new devices was a challenge but the main issue was to make all the devices and software work together and make the acquisition experience more comfortable possible. The results we obtained were quite satisfactory, we visualized the relation between the EEG and EMG signal which was our goal, but most importantly we found some practical solutions for the acquisition procedure which will help the future students to make a simpler and faster acquisition.

References

[1] Partha Mitra and Hemant Bokil in ”Observed Brain Dynamics” (2007), Chapter 7.3.4.

[2] Peter Konrad in ”The ABC of EMG A Practical Introduction to Kinesiological Electromyography” (2006)

[3] Giulia Cisotto, Anna V. Guglielmi, Leonardo Badia and Andrea Zanella in ”Classification of grasping tasks based on EEG-EMG coherence” (2018)

[4] Böcker, Koen and Avermaete, Jurgen and Berg-Lenssen in ”The international 10–20 system revisited: Cartesian and spherical co-ordinates”(1994), Pages 231-5, Volume 6

[5] NeuroImage Volume 159, 1 October 2017, Pages 403-416: Fiorenzo Artoni, Chiara Fanciullacci, Federica Bertolucci, Alessandro Panarese, Scott Makeig, Silvestro Micera and Carmelo Chisari in ”Unidirectional brain to muscle connectivity reveals motor cortex control of leg muscles during stereotyped walking” (2017)

[6] Swartz Center for Computational Neuroscience, https://sccn.ucsd.edu

[7] Makoto Miyakoshi in ”Makoto’s preprocessing pipeline”, https://sccn.ucsd.edu/wiki/Makoto’s_preprocessing_pipeline