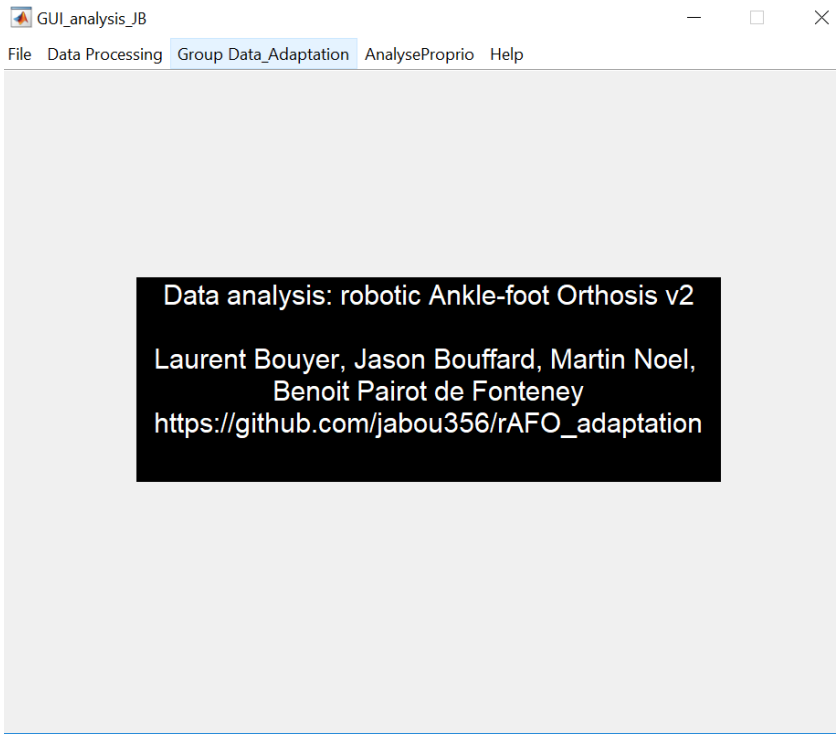
GUI_analysis_JB help file

GUI_analysis_JB has been developed by Jason Bouffard with the collaboration of Laurent Bouyer, Martin Noel (LOKOMAT) and Benoit Pairot de Fonteney between 2010 and 2018 to analyse data from the robotic ankle foot orthosis (Noel et al. 2008, IEEE Trans Neural Syst Rehabil Eng)



Here is a brief list of publications using the current or previous versions of GUI_analysis_JB, or embedded functions

Noel et al. 2008, IEEE Trans Neural Syst Rehabil Eng
Noel et al. 2009, J Neuroeng Rehabil
Blanchette et al. 2011, Gait Posture
Bouffard et al. 2014, J Neuroscience
Fournier-Belley et al. 2016, Gait Posture
Bouffard et al. 2016, Neural Plasticity
Bouffard et al. 2016, J Neurophysiol

Getting started

You can get the up-to-date version of the toolbox by downloading the latest release or cloning the master branch on: https://github.com/jabou356/rAFO_adaptation

To launch the program, just run the GUI_analysis_JB.m file. All necessary paths will be added to your Matlab Search paths. You will then be asked to open your project's parent folder. The program will load the config.mat (see Annexe 1 for the content of config file) file or create and save a default one if it doesn't exist. If you want to modify the config.mat file, just do it manually. The program brings you back to the project's parent folder and save some files in it, so organize your data structure to navigate easily between subfolders during data analysis.

Step by step guide

Process individual data files

Transfer your .raw file from WinVision into a .mat

File->Convert *.raw vers *.mat

You will have to select the *.raw file you want to convert, then save it as a *.mat file. Repeat for each file.

Output: Inputfile.mat
data(1:number of channels,1:number of samples) matrix with unfiltered data.

Combine all .mat files for one subject into a single one

Data Processing -> Combine Files from WinVisio

This function will process data (apply filters and gains) and combine multiple files. The function will first ask you to go into your subject directory (i.e. the one with your .mat files). If you have a subject specific config.mat file in your subject directory, it will be loaded. If no such file exist, it will use the generic config.mat file loaded from the project parent directory. You should only add a subject specific config.mat file in the subject folder if it is necessary (different filter to use for this subject, different channel number, etc.).

Output: combined_data.mat:
data(1:number of channels,1:number of samples): matrix with unfiltered data
fdata(1:number of channels,1:number of samples): matrix with filtered data and appropriate gains

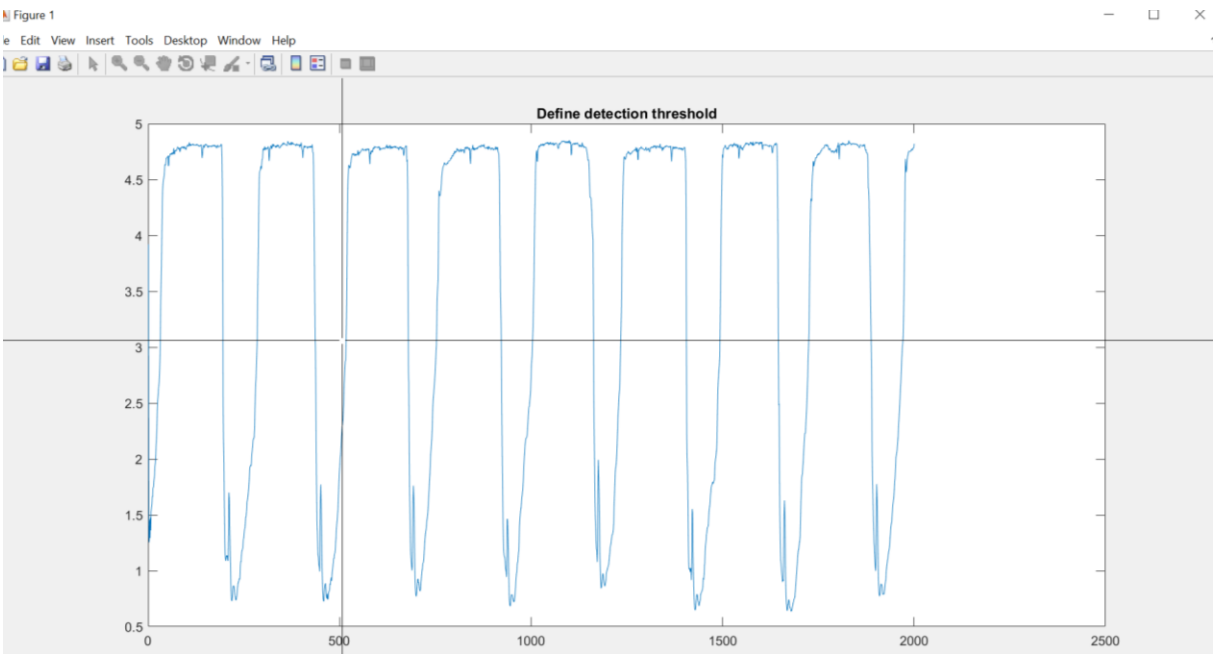
Cut file into individual strides

Data Processing -> Prepare Table -> Cut Tables Lokomat

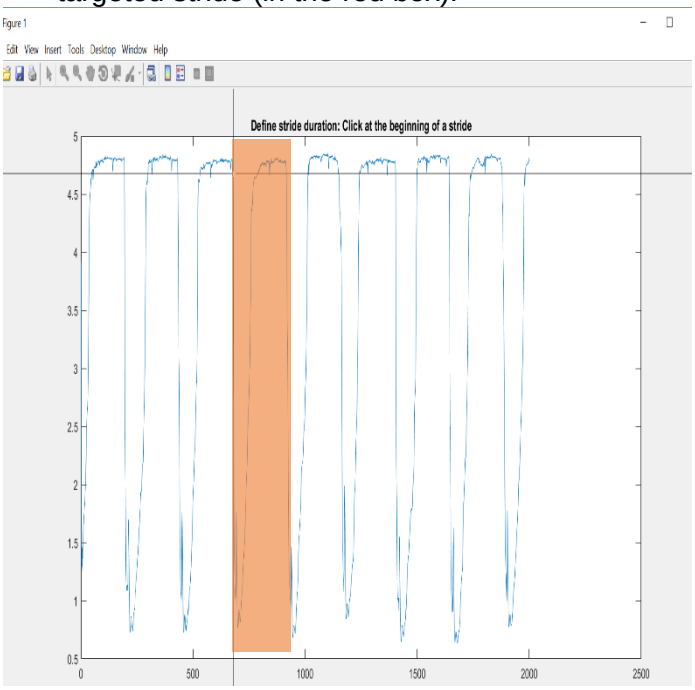
This function partition data in individual strides based on config.Sync_channel and other associated configuration parameters. It will also tag strides containing force field or reflexes (config.ISFF_channel, config.ISRFLX_channel).

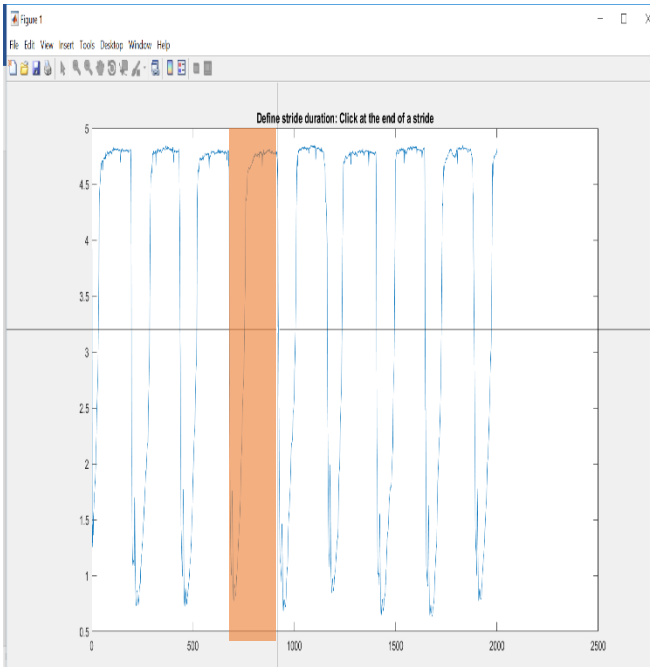
Once you activate the menu, you will have to select your subject directory. Then the function will load the combined_data.mat file and will look for a subject specific config.mat (otherwise it will take the generic configuration file).

The config.Sync_channel's signal (usually heelstrike) will be plotted and you will have to select your synchronisation threshold. An optimal threshold should be crossed once by cycle (with in the config.trig_direction).

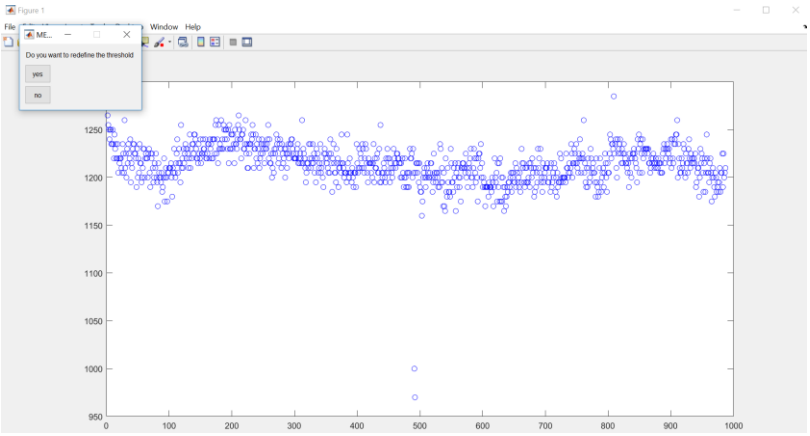


Then you will need to define an approximate stride duration by selecting the beginning and the end of a targeted stride (in the red box).



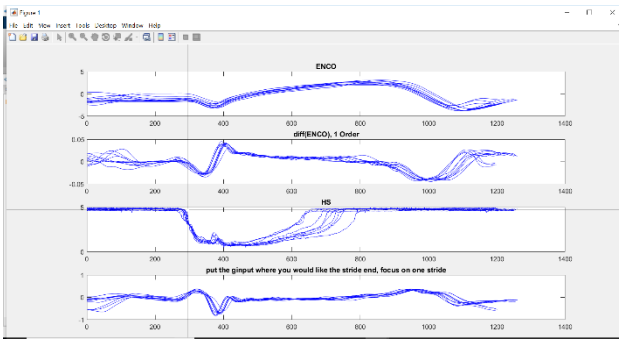


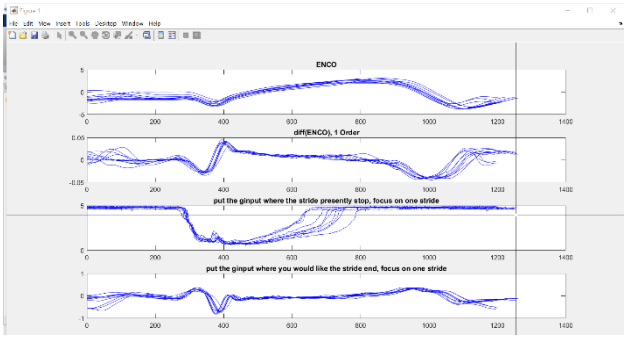
You will then be asked if you want to redefine the threshold based on a figure illustrating strides duration. If most of the strides fall within the range of expected duration, click No. If it is not the case, click on Yes and you will be asked to redefine your threshold and stride duration. This is an infinite loop. If you are not able to get a good synchronisation, just click No and modify your config.mat file to choose another Sync_channel or change other parameters (Sync_filter, Sync_diff).



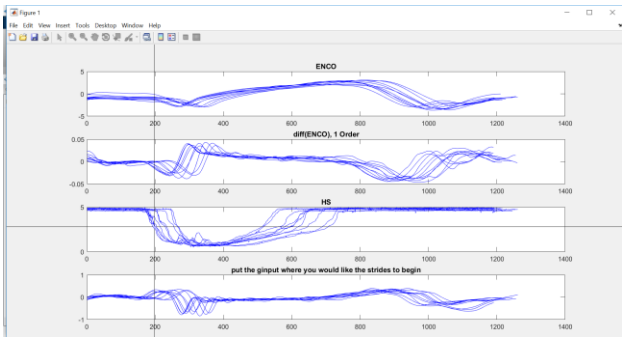
Once you accepted the data partitioning, another figure will be plotted with some signals you defined in config.setOffsetChan and config.setOffsetdiff. This step is to allow you to create an offset by choosing where you would like to see your strides ending or beginning.

If you say that you would like to choose when the strides finish, you will have to focus on a single stride, define where your would like to see it finish and where it finished in the current figure. Then, the program will shift all strides by the amount of time separating your two clicks. In the example below, I indicated that I would like to see my stride finishing at the onset of heelstike by focusing on the longest stride.

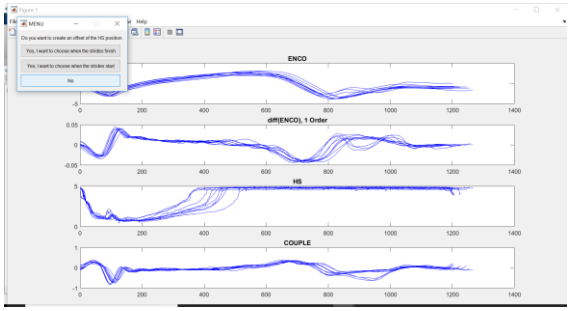




If you say that you would like to choose where the strides begin, you only need to click once, and the program will shift all strides from the selected xvalue (here around 200 ms).



In all case, you would like to get strides beginning at heelstrike (figure below. Those offsetting methods can be particularly useful when your config.Sync_channel is not a footswitch.



Output: Table_data.mat
Config : config file used to create Tables
Table: Contain partitioned data for all signals in the config.chan_name order.
Cycle_Table: Contains information about each stride.
(The structure of Table and Cycle_Table is presented in Annexe2.

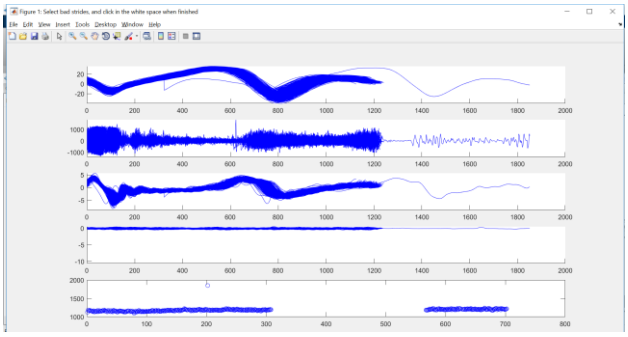
Remove bad trials

Data Processing -> validate cycles -> remove bad superpose

This function is used to visually scan the data and remove bad trials for further analyses. Bad trials are not deleted, but tagged in the 3rd row of Cycle_Table (1 = valid, 0 = non-valid).

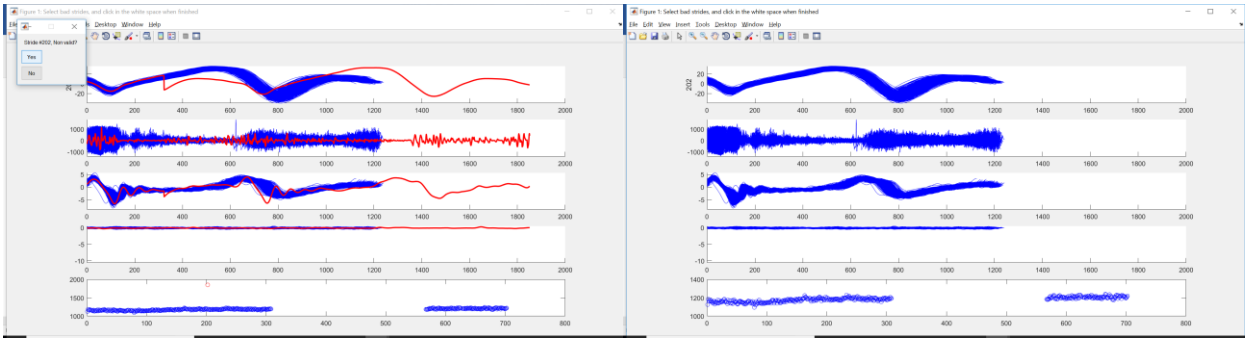
When you call the remove bad trials function in the GUI, you will be asked to indicate which signals you would like to see to validate your strides. Enter each of the signal (as named in config.chan_name) to plot with quote and between {} (e.g. {'ENCO', 'TA', 'COUPLE', 'CONS_F'}.

Then the selected channels will be plotted as well as each individual strides duration (lowest panel). First, strides without any event (Force fields or reflexes), will be presented.



From this figure, you can select a stride that is appears not valid in any of the subplots (including the lowest one). It will then be highlighted in red and you will be asked if you want to remove this stride. If you click yes, the stride will be removed from the plot and marked as non-valid in Cycle_Table.

Output: Table_data.mat
Same as Cut_table, but with Cycle_Table updated.



When you consider that all the displayed strides are valid, you click in the white space in any subplot. Strides in the next condition (e.g. force field) will then be displayed.

Motor adaptation analysis (Group files)

Combine Group Data

GroupData Adaptation-> Combine Group Data
This function combines multiple Table_data.mat files into a Group structure for the selected signals.

When you call the function, you will first have to indicate if you already have a GroupData file. If you have one, you will load it to add new subjects at the end of the file. If you don't a new file will be initialized. Then you will have to indicate which signals you would like to combine into a Group Structure. . Enter each of the signal (as named in config.chan_name) to plot with quote and between {} (e.g. {'ENCO', 'TA', 'COUPLE', 'CONS_F'}). Note that if you want to generate built in ENCO and TA variables (will be presented late), you minimally need { 'ENCO', 'TA', 'CONS_F'}.

When you are done adding subjects to the group structure, you will have to save it. Place the GroupData file in a strategic folder as new files will be saved there further in the analysis process.

Output: GroupData.mat
GroupData.Signal : *Contains all data for a given Signal.*
GroupData.Cycle_Table : Contains Cycle_Table information for all included subjects.
(The structure of GroupData file is presented in Annexe 3)

Identifications cycles

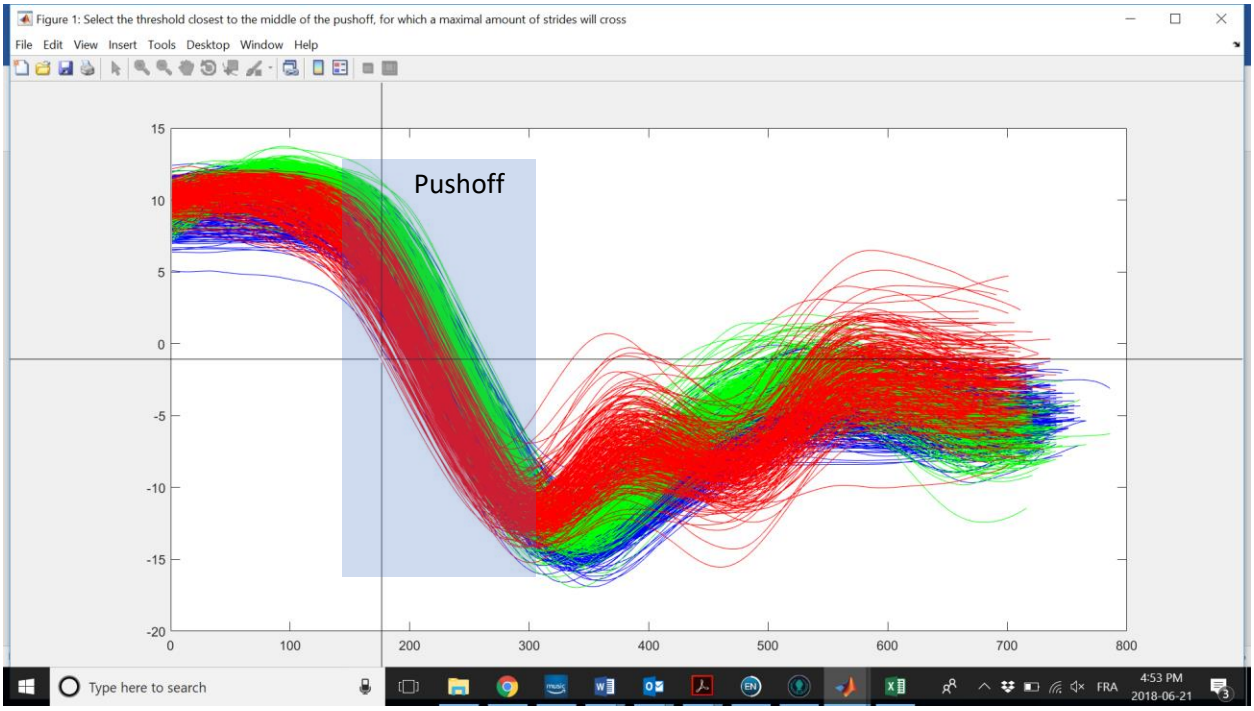
GroupData Adaptation-> Identification cycles
This function identify strides critical for the motor adaptation paradigm (Last baseline, Last force field, Last stride of the experiment.
You will just have to load your GroupData file.

Output: CyclesCritiques.mat

Synchro_pushoff

GroupData Adaptation-> Synchro_pushoff
This function create an index to isolate "the swing phase" based on the middle of the plantar flexion during pushoff. When you activate the menu, you will be asked to load your GroupData.mat file. It should be in a folder containing a ***CyclesCritiques.mat*** file *(c.f. GroupData Adaptation-> Identification cycles)*. You will next be asked if you want to use an old Sync_Data file. If you say yes, it will load the existing SyncData.mat file in your GroupData folder and add new data at the end of it.

Then, your 'ENCO' signal will be displayed. You will need to place the cursor at a y value crossed by most (almost every) strides around the middle of the pushoff phase of your stride. This step will be repeated for all subjects in your GroupData.mat file.



Output: SyncData.mat

Sync_Threshold: A vector containing the threshold for the middle of the pushoff (y value) for each subject

Sync_Timing: A cell array containing the index when the threshold was crossed for each stride of each participant (beginning of pushoff)

KinematicAnalysis, TAAnalysis, GenericAnalysis, GenericEMGAnalysis

GroupData Adaptation-> *Analysis

Those functions create outcome variables used in the different papers. In all functions, you will first be asked to load your GroupData.mat data file. Then, depending on the function you are using, you may be asked to enter the name of the signal you want to analyse. For TA analysis, there will be a menu asking you if your TA channel is named 'TA', 'RTA' or 'LTA'. For GenericAnalysis and GenericEMGAnalysis, you will be asked to write the channel name. Write it between quotes (e.g. 'COUPLE').

Finally, you will have to clean data as in *Data Processing -> validate cycles -> remove bad superpose.* This allows you to verify the quality of the synchronisation (SyncData).

Note that TA Analysis has to be performed after ENCO analysis because it uses some information generated during this step (e.g. baseline strides used for ENCO.baseline).

For ENCO Analysis, you have to include CONS_F in your GroupData file before generating CyclesCritiques and SyncData.

Output: AnalENCO, AnalTA or AnalSignal
The structure of those files are presented in Annexe 3.

Proprioception analysis (single subject)

Proprioception analysis

AnalyseProprio-> AnalProprio

This function generate all variable used to compute Proprioceptive thresholds and uncertainty regions. The input is a validated Table_data.mat *(Data Processing computed).* When you call the function, you will be asked

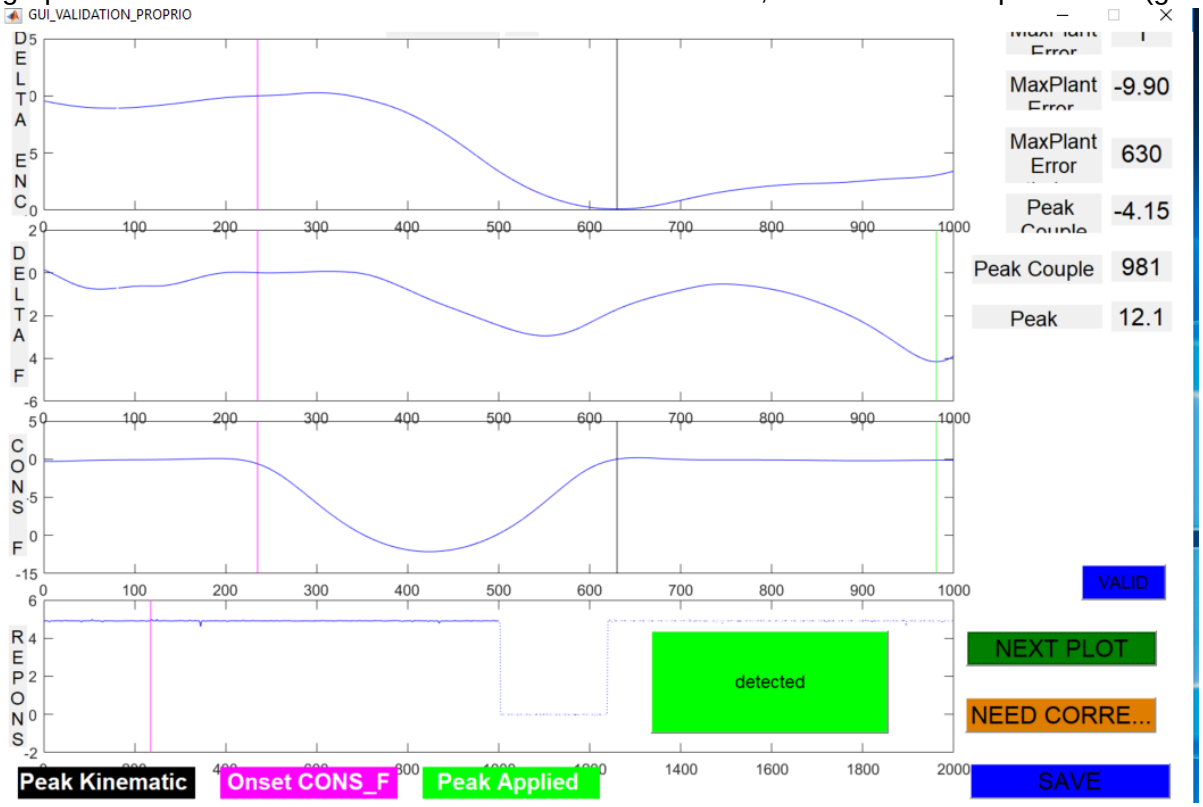
the direction of the force field during the proprioception task: toward plantarflexion, resist dorsiflexion OR toward dorsiflexion, resist plantarflexion. Then you will have to select the threshold to synchronize your data on the middle of the pushoff to isolate swing *(see GroupData Adaptation-> Synchro_pushoff).* Finally, you will clean data to verify synchronisation quality *(see Data Processing -> validate cycles -> remove bad superpose).*

Output: AnalProprio.mat
The structure of those files are presented in Annexe 4.

Validate proprioception analysis

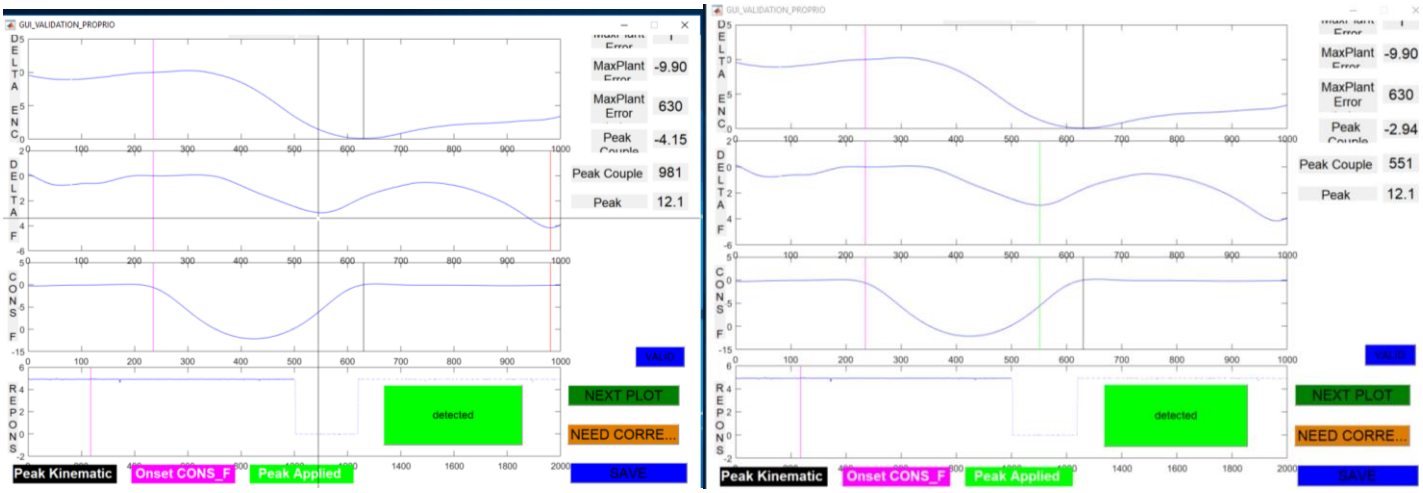
AnalyseProprio-> ValidationProprio

This function allow you to validate the different variables relevant to Proprioception Analysis using the following graphical user interface. To show the first stride to validate, activate the next plot button (green).



Upper panel: Kinematic error signal, Panel 2: Applied torque (after removing baseline torque), Panel 3: Force command, Lower panel: Participant's response (button). Note that for the lower panel (button), the current (full line) and the next (dashed line) strides are illustrated. The vertical lines indicate the timing of the following events: Onset of Force command, Peak kinematic error, Peak applied torque.

You mostly need to verify the timing of the events with this GUI. You would expect that the peak kinematic error and peak applied torque occur slightly after the peak force command. In the illustrated example, the peak kinematic error is adequate. However, the peak applied torque is too late and do not illustrate the effects of the force field. To correct this, you need to activate the need correction (orange) button. Then you will need to click on the vertical line representing the event you want to correct (green line for the peak torque). Once you selected the event to correct, the line will become red and as cursor will appear. Place it where the event should be.



Note that the values of Peak COUPLE and Peak COUPLE timing have been updated on the upper right part of the GUI.

You can also switch a detected stride to a non-detected stride by activating on the big green button (red if non-detected) in the lower panel. You may want to do this if the participant pressed the button before the force was actually applied. Finally, you can switch a stride to non valid by clicking on the Blue “valid” button. Once activated, this button will become red and all values will be switched to NaN. When you are happy with the displayed stride, click on Next Plot. Repeat the procedure for all strides. When you are done, activate the Save button.

Output: AnalProprio.mat (updated)
The structure of those files are presented in Annexe 4.

Generate Proprioception outcome measures.

AnalyseProprio-> ProprioOutcome
This function generates outcome measures for proprioception analysis (threshold and uncertainty based on Kinematic error and applied force. ADD FIGURES DETAILS HERE

Output: AnalProprio.mat (updated with outcome measures.)
The structure of those files are presented in Annexe 4.

Annexe 1: config.mat

Channel identifier (all vectors, and cell array {chan names})

```
config.EMG_channels = [1,2,3,4,5]; %Original EMG chan number in WinVisio

config.Other_channels = [8,14,15,16]; % Original chan number (not EMG, not squared) in WinVisio

config.Square_channels = [10, 12]; % Original chan number (square waves) in WinVisio

config.chan_name = {'TA','SOL','GM','VL','RF','Knee','CONS_F',...
    'ENCO','COUPLE','Response','HS'}; %chan_name:EMG goes first, then other_channels, then square channels
```

Identifier event channels with properties (all scalars)

```
config.ISFF_channel = 7; % Channel with FF command (e.g. CONS_F)

config.FFdetect_level = 0.6; % Threshold for FF detection (should be an positive scalar)
```

config.ISRFLX_channel = 0; %Channel to identify RFLX, catch, anticatch (e.g. Memory gate)

config.RFLXdetect_level = 0.4; % Threshold for event channel (should be a positive scalar)

Partition table into individual strides (all scalars or char)

config.Sync_channel = 11; % Channel # used to partition data

config.trig_direction = '<'; %Direction of the Sync_channel on which you want to trigger '<': descending, '>': ascending

config.trig_diff = 0; % set to diff order of sync chan (0 if no differentiation)

config.trig_lowpass = 0; %Set low pass filter frequency of sync chan (0 if no filter)

config.trig_Nlowpass = 0; % Set low pass order if to be used

config.pct_refractaire = 0.8; % Duration of refractory period (must be between 0 and 1)

Preprocessing properties

config.chan_gain = [250, 250, 250, 250, 250, 90, 4, 8, 10, 1, 1]; % vector with chan gains

config.EMG_filter = [20 450]; %:EMG :[low pass Fz, High pass Fz]

config.Other_filter = 15; % Other channles:[Low pass Fz]

config.Order = 2; %filter order (divide the wanted number by two as we use filtfilt

config.sFz = 1000; %Sampling frequency (scalar)

Config for Adaptation Analyses

config.useSync = 1; %useSync:1: Use sync data for analyses (e.g. middle pushoff), 0:start at first data point of each stride

Config for Proprio Analysis

config.AnkleName = 'ENCO'; % Name of Ankle signal for proprio

config.COUPLEName = 'COUPLE'; % Name of COUPLE signal for proprio

config.ResponseName = 'Response'; % Name of Response signal for proprio

config.ForcecommandName = 'CONS_F'; % Name of Force command signal for proprio

config.ResponseTh = 1.5; % Threshold for the detection of the button