

# I-Spin Quattrocento: a step-by-step user manual

## **Introduction:**

I-spin Quattrocento is a Matlab app interfacing with a multichannel EMG recording system (i.e., EMG-Quattrocento, 400 channel EMG amplifier; OT Bioelettronica, Italy) to identify the firing activity of single motor units in real-time and translate it into several forms of visual feedback. Users can implement this tool during experiments requiring the control of the firing activity of individual motor units or pairs of motor units in real-time.

We provide here a step-by-step protocol to facilitate the implementation of I-Spin live in any experimental settings. You can also read our preprint that describes the method, the main steps of the experiments, and the capabilities of the app (<https://www.biorxiv.org/content/10.1101/2023.04.14.536933v2>).

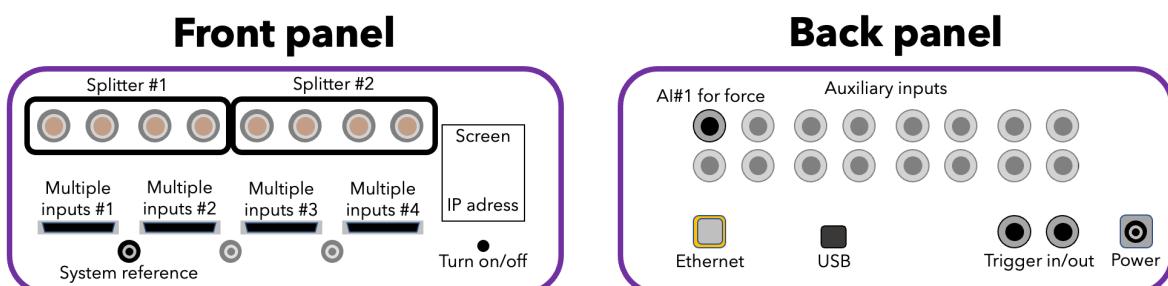
## **Abstract of the preprint:**

Decoding the activity of individual neural cells during natural behaviours allows neuroscientists to study how the nervous system generates and controls movements. Contrary to other neural cells, the activity of spinal motor neurons can be determined non-invasively (or minimally invasively) from the decomposition of electromyographic (EMG) signals into motor unit firing activities. For some interfacing and neuro-feedback investigations, EMG decomposition needs to be performed in real-time. Here, we introduce an open-source software that performs real-time decoding of motor neurons using a blind-source separation approach for multichannel EMG signal processing. Separation vectors (motor unit filters) are optimised for each motor unit from baseline contractions and then re-applied in real-time during test contractions. In this way, the firing activity of multiple motor neurons can be provided through different forms of visual feedback. We provide a complete framework with guidelines and examples of recordings to guide researchers who aim to study movement control at the motor neuron level. We first validated the software with synthetic EMG signals generated during a range of isometric contraction patterns. We then tested the software on data collected using either surface or intramuscular electrode arrays from five lower limb muscles (gastrocnemius lateralis and medialis, vastus lateralis and medialis, and tibialis anterior). We assessed how the muscle or variation of contraction intensity between the baseline contraction and the test contraction impacted the accuracy of the real-time decomposition. This open-source software provides a set of tools for neuroscientists to design experimental paradigms where participants can receive real-time feedback on the output of the spinal cord circuits.

## **Before starting the experiment:**

### Hardware requirements.

I-Spin Quattrocento directly interfaces with a multichannel EMG recording system developed by OT Bioelettronica, the Quattrocento. This system allows you to simultaneously record data from six arrays of surface or intramuscular electrodes.

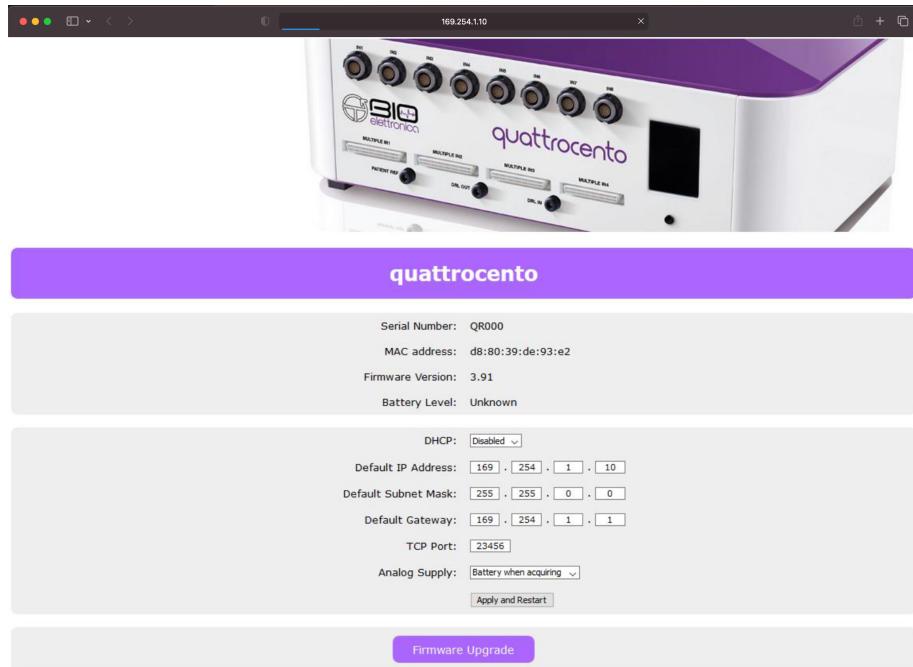


**Figure 1.** Views of front and back panels of the Quattrocento. Connect the arrays to the splitters or multiple inputs 1 to 4. Connect the system to the participant using a wet band and the patient reference cable. Connect the dynamometer to auxiliary input #1 using a BNC cable. Before starting an experiment: i) connect the computer and the quattrocento using an ethernet cable, ii) connect the quattrocento to the power and turn it on, and iii) check the connection in the web browser using the IP address displayed on the screen.

For this, you will need:

- An ethernet cable to connect the Quattrocento to the computer.
- Up to six arrays of 64 surface electrodes (e.g., GR04MM1305, GR08MM1305, GR10MM0808) with the associated adhesive foams to stick the arrays over the skin.
- Up to six pre-amplifiers
- Up to six cables to connect the pre-amplifiers to Splitters #1, #2 or Multiple Inputs #1 (MI#1), MI#2, MI#3, and MI#4 on the front panel of the Quattrocento.
- A system recording the joint force or torque (e.g., an isokinetic dynamometer) with an BNC cable going to the Auxiliary Input 1 on the back panel of the Quattrocento.
- A patient reference cable with a band dampened with water.
- Stackable cables connecting each of the pre-amplifiers to another band dampened with water.

Connect the computer and the Quattrocento using the ethernet cable first. Then, connect the Quattrocento to the power and turn it on using the black button below the small screen. An IP address should appear. Before starting the experiment, you should check the connection between the computer and the Quattrocento by opening a web browser and typing the IP address in the search bar as follow: <http://169.254.1.10/>.



**Figure 2.** Webpage showing the parameters of the Quattrocento. If this page does not appear, i) turn off and unplug the Quattrocento from power supply, ii) reconnect the ethernet cable, iii) plug the Quattrocento to power and turn it on.

### Software requirements.

I-Spin live works on any modern computer running Matlab. The current version of I-Spin live has been developed on Matlab R2022b and tested on a laptop equipped with an Apple M1 Max chip and 64 GB of RAM. However, we successfully ran I-Spin live on multiple Windows and MacOs computers with versions of Matlab ranging from 2018a to 2023a. The refresh rate of the screen can impact the visualisation of EMG channels. It is recommended to display the app to participants on a large screen with a reasonable frame rate (e.g., 60 Hz).

I-Spin live has three dependencies: the signal processing toolbox, the image processing toolbox, and the statistics and machine learning toolbox. Install them before running the app for the first time.

### I-Spin live installation.

Up-to-date versions of I-Spin live are uploaded on GitHub in the following repository <https://github.com/simonavrillon/I-Spin>. The repository is structured with one folder 'lib' containing all the functions needed to run I-Spin\_Quattrocento, and three scripts containing either the full code (ISpin\_Quattrocento\_exported.m), the code + the design of the app (ISpin\_Quattrocento.mlapp), an installation file to have the app directly runnable from Matlab's app library (ISpin\_Quattrocento.mlappinstall).

#### *Option #1: ISpin\_Quattrocento\_exported.m:*

To run ISpin\_Quattrocento\_exported.m, you first need to add the full folder with the library of functions and the main code to Matlab's path. i) Go to the 'Home' tab,

'Environment' table, and click on 'Set Path'; ii) Click on 'Add with Subfolders...', find the folder with all the scripts, and click on 'Open'; iii) Click on 'Save', go to the 'Editor' tab, click on 'Open', find the script ISpin\_Quattrocento\_exported.m, and click on 'Run'.

#### *Option #2: ISpin\_Quattrocento.mlapp*

To run ISpin\_Quattrocento.mlapp, you need to open the app editor. i) Go to the 'Apps' tab and click on 'Design App'; ii) Click on 'Open', find the script ISpin\_Quattrocento.mlapp and open it; iii) Go to the 'Designer' tab and click on 'Run'.

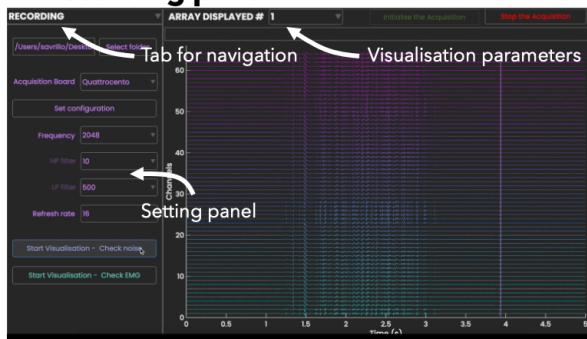
#### *Option #3: ISpin\_Quattrocento.mlappinstall*

Before the first use, you need to install the I-Spin live app. Open the script ISpin\_Quattrocento.mlappinstall and click on 'Install' in the pop-up window. For all the following uses, open Matlab, go to the 'Apps' tab, find the app in the apps library and click on the shortcut to run the app.

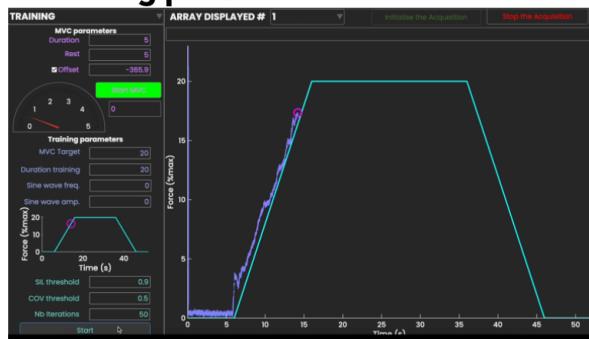
### **Description of the user interface:**

I-Spin live has three main panels for the successive steps preceding the real-time decoding of the firing activity of individual motor units and biofeedback experiments: 'Recording', 'Training', and 'Manual Editing'. On the last panel, 'Biofeedback', you can display to the participants three forms of visual feedback: i) a raster plot of the firing times of all the motor units from one array; ii) a quadrant with a cursor moving according to the firing activity of two single motor units; iii) a scrolling path with the smoothed firing activity of a motor unit and a scrolling path to track. You can navigate between panels using the tabs on the top-left corner.

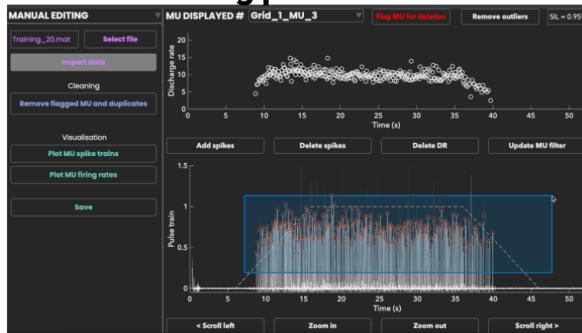
#### **1. Recording panel**



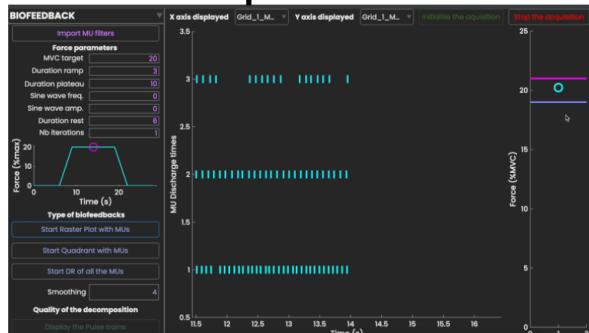
#### **2. Training panel**



#### **3. Manual Editing panel**



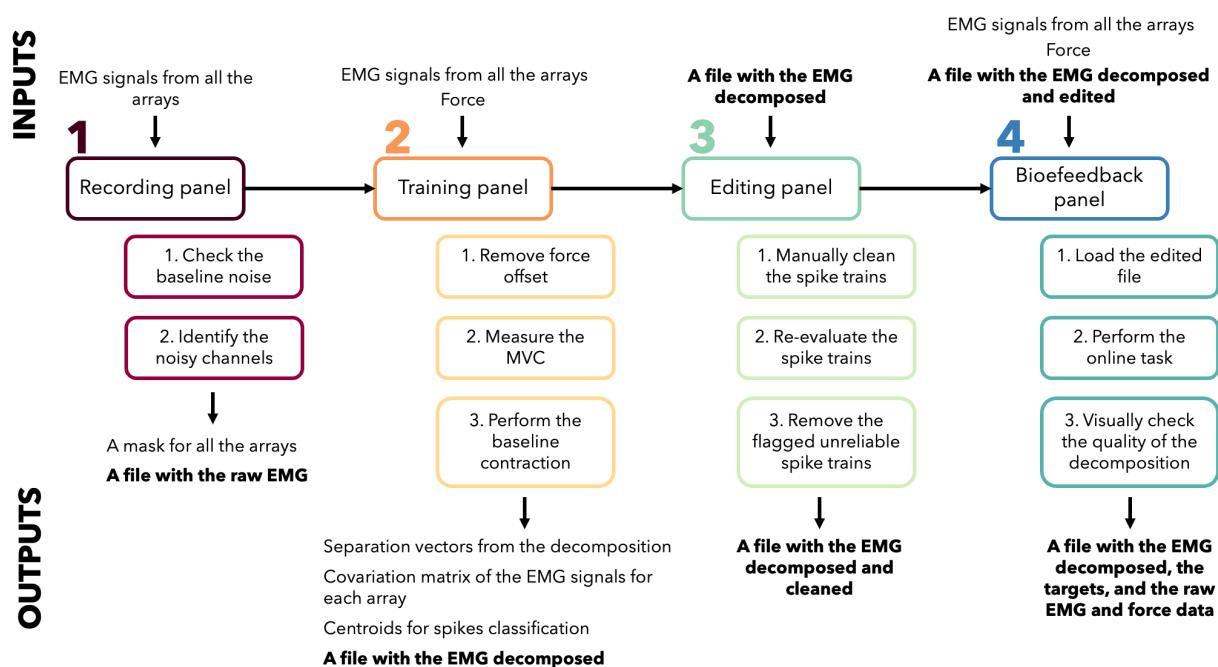
#### **4. Biofeedback panel**



**Figure 3.** Overview of the panels from the I-Spin live app.

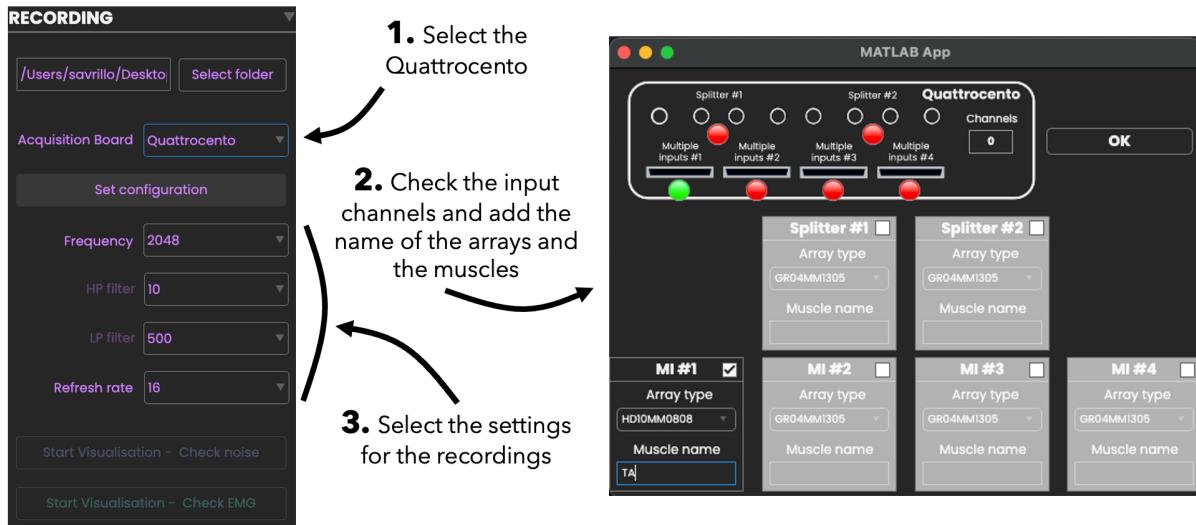
## **Workflow of an experimental session:**

A typical session starts with the visualisation of raw EMG signals to check the baseline noise and remove the channels with artifacts and low signal-to-noise ratio. This step generates a mask, which is applied over the EMG data for the rest of the session to remove noisy channels. Second, the offset of the auxiliary channel recording the force is measured and removed for the rest of the session. Then, the participant performs a series of maximal voluntary contractions to measure the maximal force output and provide a target in % of maximal force during the subsequent contractions. Third, the participant performs a submaximal trapezoidal isometric contraction by tracking a visual target. EMG signals collected during this contraction are decomposed using a blind-source separation algorithm to identify a sample of motor units. These motor units are then used for all the forms of biofeedback in real-time. The users can visualise and manually edit the identified motor units to improve the performance of the decomposition in real-time. Finally, the participant performs a series of tasks in real-time using one of the three forms of visual feedback.



**Figure 4.** Workflow of a typical experimental session with I-Spin live.

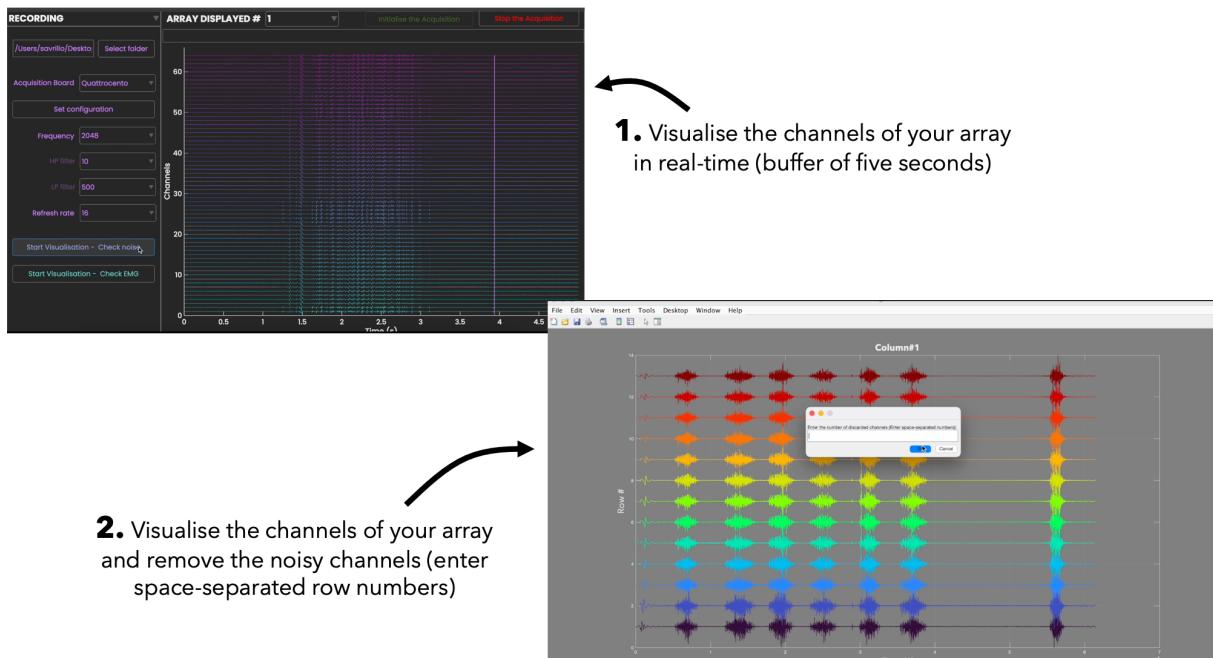
## Recording panel.



Select a folder where all the files generated during the session will be saved. Then select Quattrocento in the dropdown list of acquisition boards and set the configuration. For this, select the input channels in the pop-up window by ticking the boxes in upper right corners. Add the names of the arrays and muscles and click on 'OK'. Once all the settings are set, click on the button '[Initialise the Acquisition](#)'. It will generate a series of bytes sent to the EMG recording system to set the recording parameters. **You will click on this button before each recording to update the series of bytes.** After this, the buttons 'Start Visualisation - Check Noise' and 'Start Visualisation - Check EMG' are available.

'Start Visualisation - Check Noise' starts the recording of 15 seconds of EMG data during which the participant should rest. At the end of the 15 seconds, a bar plot appears displaying the root-mean-squared amplitude for each of the channels. You can repeat this step until all the channels have a low level of noise or go to the next step to add a mask over the noisy channels.

'Start Visualisation - Check EMG' starts a recording of 30 seconds during which the participant should perform submaximal contraction at random levels of force to visually estimate the signal-to-noise ratio. At the end on the 30 seconds, a plot shows the raw EMG traces for all the channels of each array, and a pop-up window allows you to select the noisy channels. It will generate a mask and a file with the raw EMG signals.

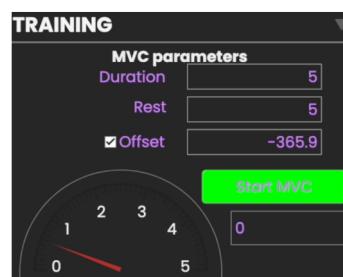


At the end of this step, a file named 'EMGchecking...' is saved in the folder.

### **Training panel.**

Before removing the offset and starting the MVC, click on the button 'Initialise the Acquisition'.

- 2.** Check the box for recording and removing the force offset



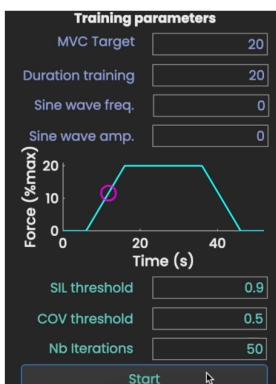
- 1.** Set the parameters for the maximal voluntary contractions. The participant will perform three maximal voluntary contractions

- 3.** Start the MVC. The button turns green when the participant must perform the MVC

At the end of this step, a file named 'MVC...' is saved in the folder.

**2.** Set the parameters for the decomposition

- The SIL threshold enables you to remove the spike trains with a short distance between the spikes and the noise
- The COV threshold enables you to remove the spike trains with a high variability between interspike intervals
- The number of iterations enables you to potentially increase the number of identified motor units



**1.** Set the parameters for the baseline contraction. The target is automatically updated.

Before clicking on the button 'Start', click on the button '[Initialise the Acquisition](#)'.

The participant will then track a target with a cursor moving in the vertical direction according to the level of force produced onto the dynamometer. At the end of this step, the decomposition starts with a pop-up window displaying a waiting bar. At the end of the decomposition, a file named 'Training...' is saved in the folder.

Note that you can edit some parameters within the code to optimise the decomposition (*ISpin\_Quattrocento\_exported.m* or *ISpin\_Quattrocento.mlapp*). Specifically, you can change the number of channels generated by the signal extension function in the following line:

```
nbextchan = 1000;
```

and the contrast function used in the fixed-point algorithm in the following lines:

```
maxiter = 500; % max number of iterations for the fixed point algorithm
w = fixedpointalg(w, X, B, maxiter, 'skew');
```

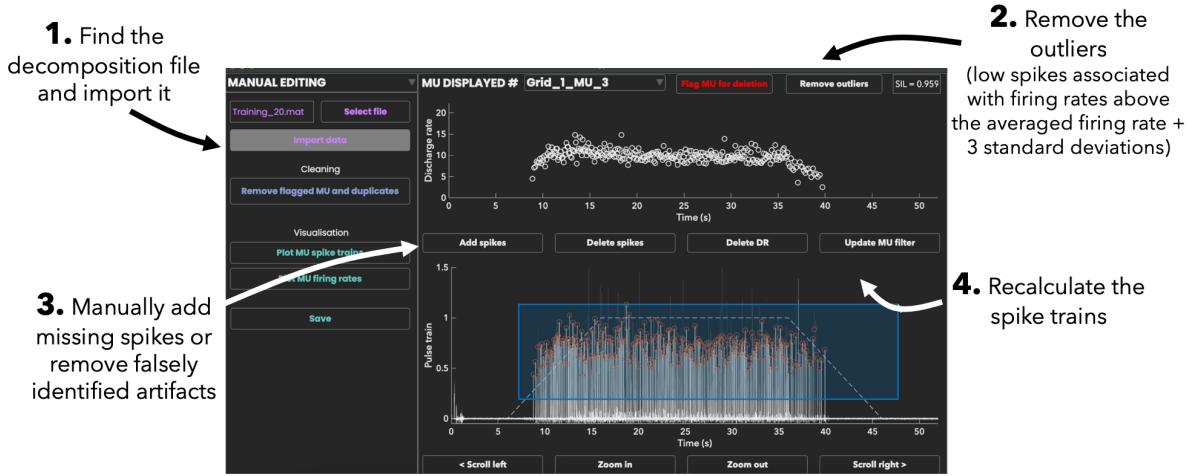
The contrast functions '**skew**', '**kurtosis**', and '**logcosh**' are available in *fixedpointalg.m*. We set the number of extended channels to 1000 and selected the contrast function '**skew**' in the associated preprint.

#### Manual Editing panel.

The manual editing step consists of i) removing spikes causing erroneous discharge rates (outliers), ii) adding discharge times clearly separated from the noise, iii) recalculating the separation vector, iv) reapplying the separation vector on the entire EMG signals, and v) repeating this procedure until the selection of all the discharge times appears to have been achieved.

The impact of manual editing on the decomposition can be observed in the associated preprint in Figure 2. For additional information about the full process to edit motor unit spike trains, please read the following paper.

Avrillon, S., Hug, F., Baker, S.N., Gibbs, C., and Farina, D. (2024). Tutorial on MUedit: An open-source software for identifying and analysing the discharge timing of motor units from electromyographic signals. *J Electromyogr Kinesiol* 77, 102886. 10.1016/j.jelekin.2024.102886.

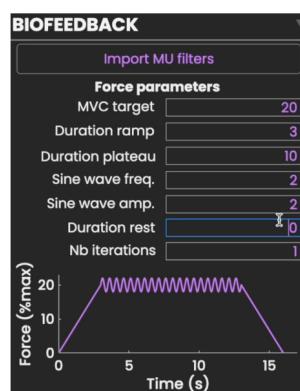


**5.** If the spike train is unreliable, flag the motor unit

Once the editing of all the identified motor unit spike train is completed, i) click on the button ‘Remove flagged MU and duplicates’ to only keep the reliable and unique motor units and ii) click on ‘Save’ to save a file named ‘Training...\_edited’ with the edited spike trains.

### Biofeedback panel.

1. Import the edited motor unit spike trains to update the decomposition parameters

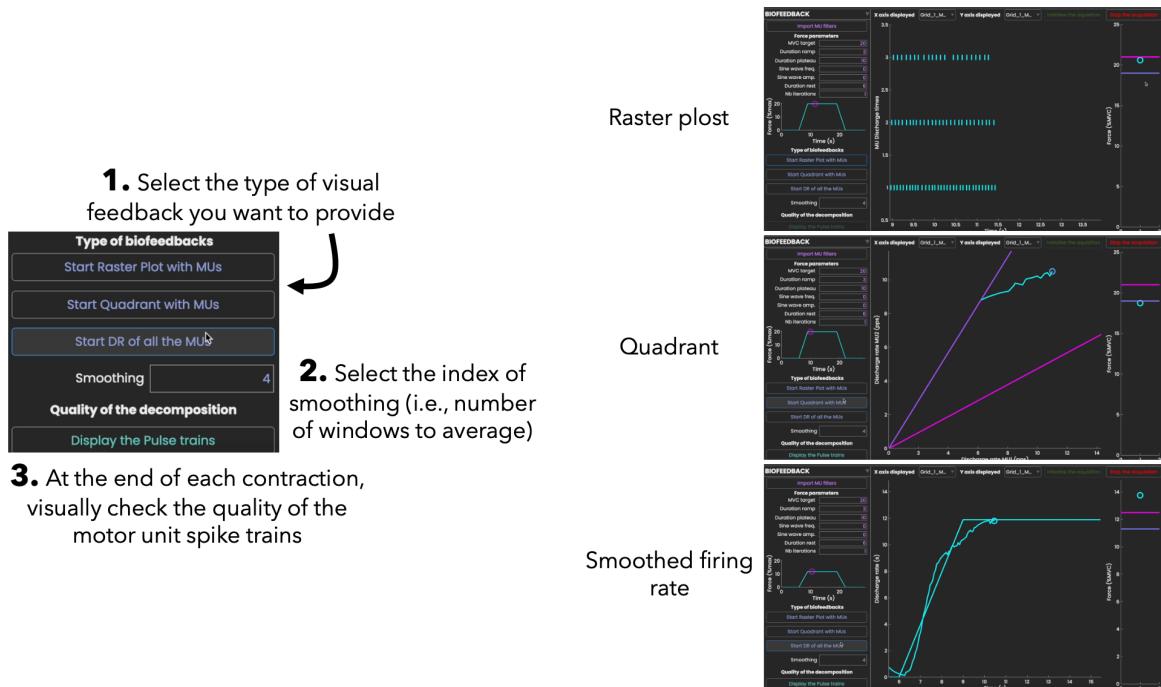


2. Set the parameters for the online contractions. The target is automatically updated.

Before clicking on the button ‘Start’ for each form of visual feedback, click on the button ‘Initialise the Acquisition’.

On the top of the main panel displaying the visual feedback, you can choose the motor units to display. For the raster plot and the scrolling path with the smoothed firing rate of a motor unit, the decomposition is performed on the array displayed in the dropdown menu ‘x axis...’. For the quadrant with a cursor moving according to

the activity of two single motor units, the decomposition is performed on the arrays displayed in the dropdown menus 'x axis...' and 'y axis...'!



At the end of each recording, a file named 'Online...' is saved in the folder.

### **Citation and technical support.**

For technical assistance and support, please contact:

Dr. Simon Avrillon

Sir Michael Uren Hub

Imperial College London

86 Wood Ln

London W12 0BZ

E-mail address: [s.avrillon@imperial.ac.uk](mailto:s.avrillon@imperial.ac.uk)

ORCID: 0000-0002-2226-3528

If you use I-Spin live in your experimental setting, please cite the following preprint.

Rossato J, Hug F, Tucker K, Gibbs C, Lacourpaille L, Farina D & Avrillon S. (2024). I-Spin live: An open-source software based on blind-source separation for real-time decoding of motor unit activity in humans. *bioRxiv*.