I-Spin live: a step-by-step user manual

Introduction:

I-SpinSAGA live is a MATLAB application for the acquisition and analysis of HD-EMG signals that interfaces with SAGA, integrating the SAGA Interface for MATLAB. I-Spin offers the possibility to identify discharge activity of single motor units in real-time. It has the capability to use the learned motor unit filter in real-time, which opens up the possibility to supply subjects with visual feedback on the Motor Unit activity during experiments.

I-SpinSAGA is an adaptation from I-Spin live which has been developed by J. Rossato et al. (please cite https://doi.org/10.7554/eLife.88670.1 when you use the library for your experiments). More detailed information on the implementation and available procedures can be found on the GitHub page https://github.com/JulR14/I-SpinSAGA of I-SpinSAGA. Here, you can also find a step-by-step protocol that describes the application in detail.

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 discharge 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 spinal motor neurons using a blind-source separation approach for multichannel EMG signal processing. Separation vectors (motor unit filters) are identified for each motor unit from a baseline contraction and then re-applied in real-time during test contractions. In this way, the discharge activity of multiple motor units can be provided as visual feedback in real-time. 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 tested the software on data collected using either grids of surface electrodes 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 interface 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.

The current version of I-Spin live can directly interface with a multichannel EMG recording system developed by TMSi, SAGA. This system allows you to simultaneously record data from two grids of surface electrodes.

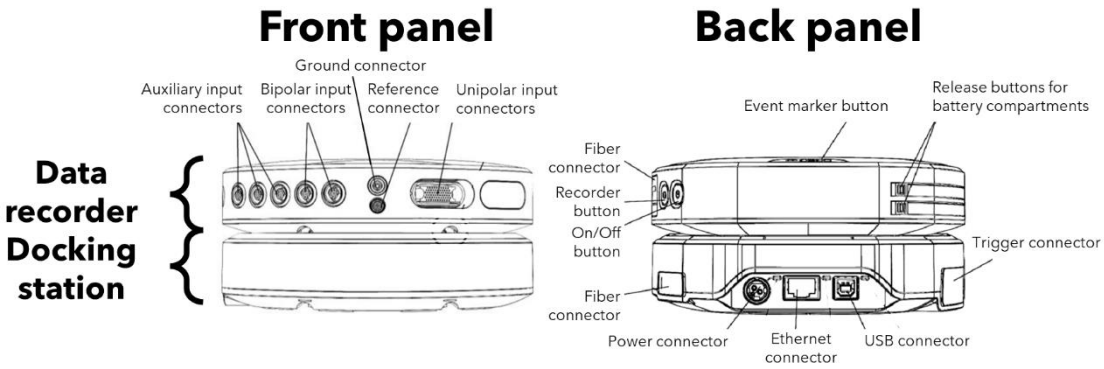


Figure 1. Views of front and back panels of SAGA. Connect the grids to Unipolar inputs. Connect the system to the participant using a wet band and the patient ground cable and an electrode and the patient reference cable. Connect the dynamometer to auxiliary input #1 using an appropriate cable. Before starting an experiment: i) connect the computer and SAGA using an USB cable, ii) connect SAGA to the power and turn it on.

For this, you will need:

- An USB cable to connect SAGA to the computer.
- Up to two grids of 32 surface electrodes or one grid of 64 surface electrodes with the associated adhesive foams to stick the grids over the skin.
- Up to two cables to connect the grids to unipolar connectors on the front panel of SAGA.
- A system recording the joint force or torque (e.g., an isokinetic dynamometer) with an appropriate cable going to the Auxiliary Input connector 1 on the front panel of SAGA.
- A patient ground cable with a band dampened with (saline) water.
- A patient reference cable with an electrode.

Connect the computer and SAGA using the USB cable first. Then, connect SAGA to the power and turn it on. Before starting the experiment, you should ensure that the appropriate SAGA driver is installed in your computer (it can be downloaded from: https://info.tmsi.com/software-tmsi-saga-device-driver-windows-v2.0).

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 i7-10750H CPU and 32 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 four dependencies: the signal processing toolbox, the image processing toolbox, and the statistics, machine learning toolbox and SAGA interface toolbox. Install them before running the app for the first time. The SAGA interface toolbox needs to be downloaded from TMSi website (https://info.tmsi.com/download-tmsi-matlab-interface-saga-v2.0-windows-os).

I-Spin live installation.

Up-to-date versions of I-SpinSAGA live are uploaded on GitHub in the following repository https://github.com/JulR14/I-SpinSAGA. The repository is structured with one folder 'lib' containing all the functions needed to run I-Spin live, one folder '+TMSiSAGA' containing all the functions needed to interface with SAGA (SAGA interface toolbox, which needs to be downloaded from TMSi website and copied within the repository), and three scripts containing either the full code (ISpin_exported.m), the code + the design of the app (ISpin.mlapp), an installation file to have the app directly runnable from Matlab's app library (ISpin.mlappinstall).

Option #1: ISpin_exported.m:

To run ISpin_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_exported.m, and click on 'Run'.

Option #2: ISpin.mlapp

To run ISpin.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.mlapp and open it; iii) Go to the 'Designer' tab and click on 'Run'.

Option #3: ISpin.mlappinstall

Before the first use, you need to install the I-Spin live app. Open the script ISpin.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 discharge activity of individual motor units and biofeedback experiments: 'Recording', 'Training', and 'Edition'. On the last panel, 'Bioefeedback', you can display to the participants three forms of visual feedback: i) a raster plot of the discharge times of all the motor units from one grid; ii) a quadrant with a cursor moving according to the discharge activity of two single motor units or the discharge activity of two populations of motor units; iii) a scrolling

path with the smoothed discharge activity of all the motor units from one grid. You can navigate between panels using the tabs on the top-left corner.

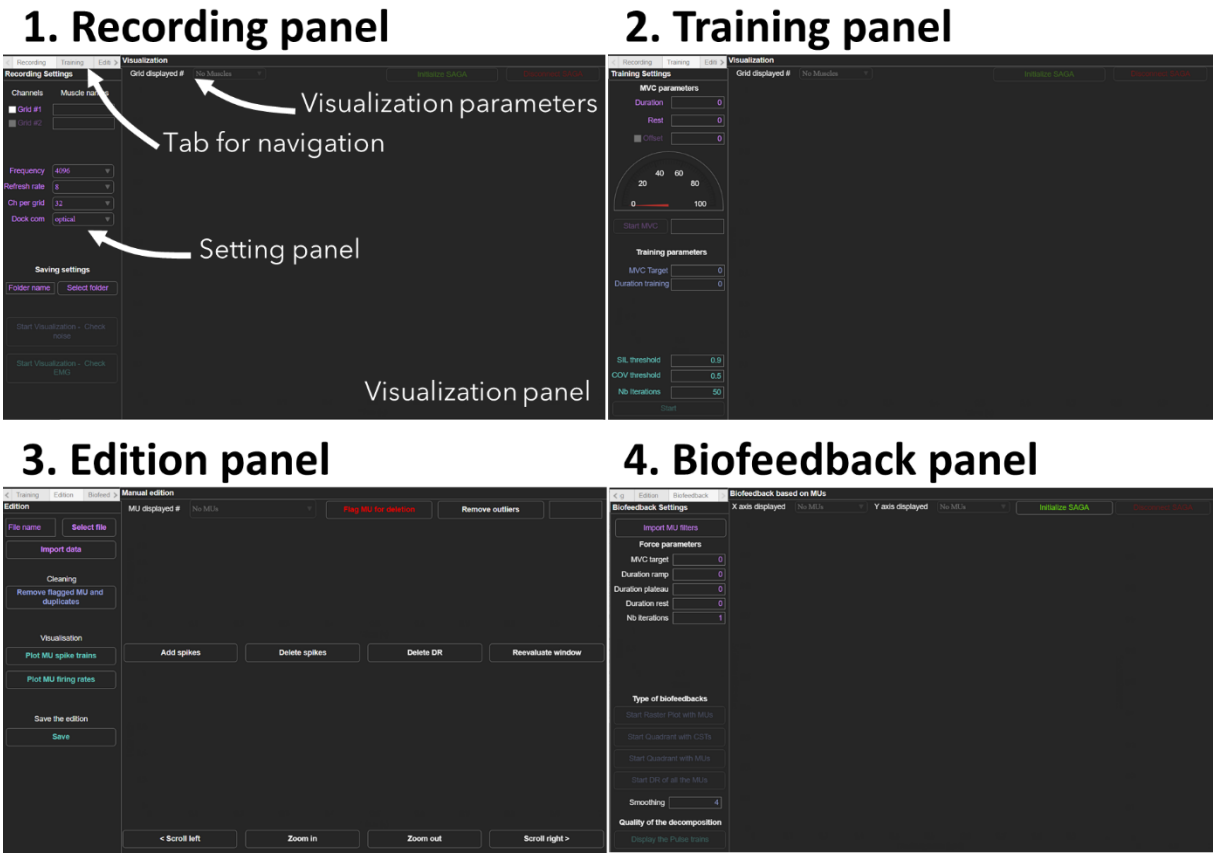


Figure 3. Overview of the panels from the I-Spin live app.

<u>Workflow of an experimental session:</u>

A typical session starts with the visualization 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 successive tasks. 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 visualize 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.

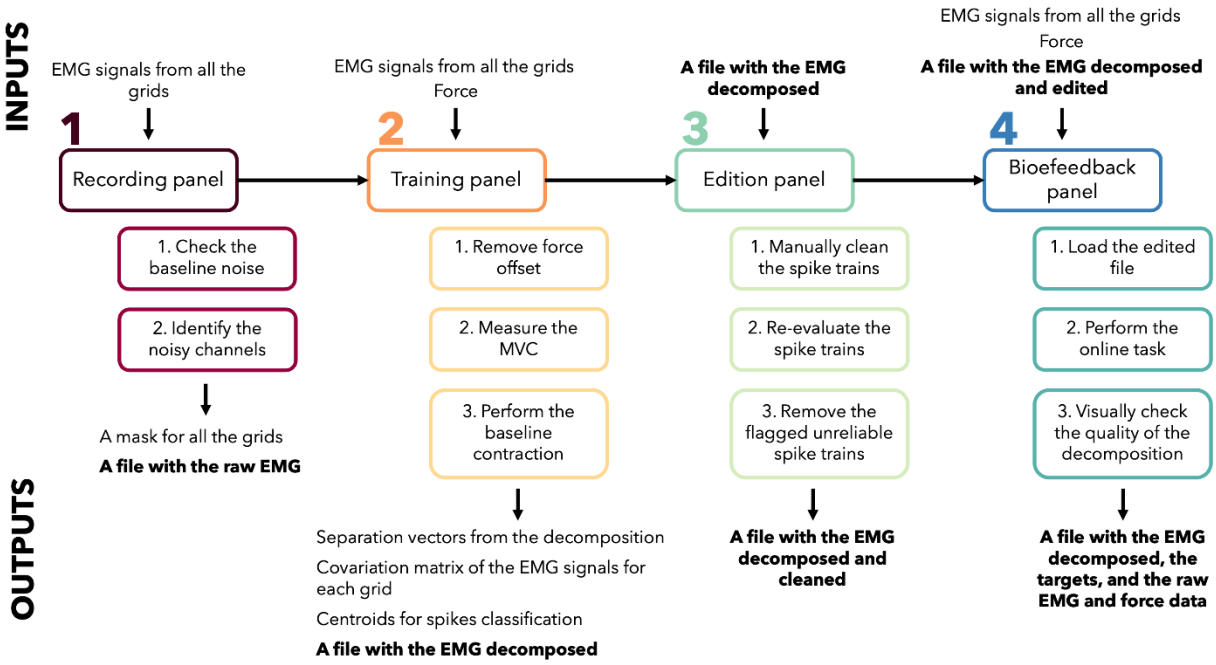
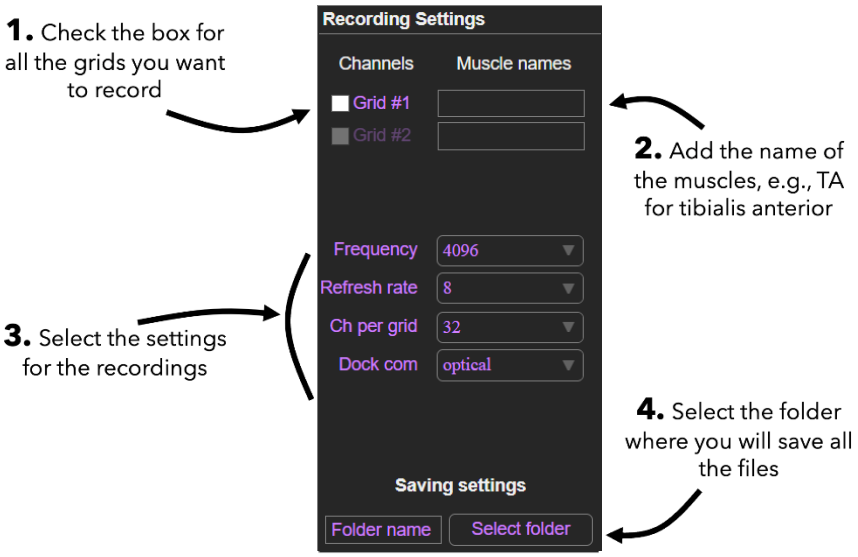


Figure 4. Workflow of a typical experimental session with I-Spin live.

Recording panel.

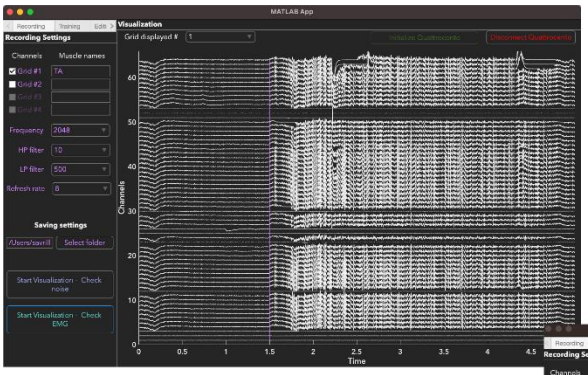


Once all the settings are set, click on the button 'Initialize SAGA'. It will start communication with the EMG recording system to set the recording parameters. After this, the buttons 'Start Visualization – Check Noise' and 'Start Visualization – Check EMG' are available.

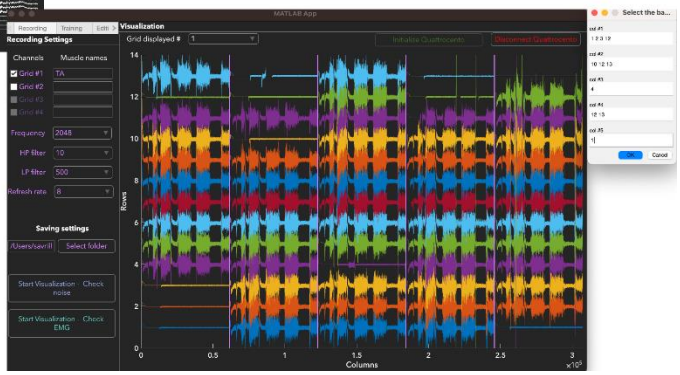
'Start Visualization – Check Noise' starts the recording of 15 seconds of data during which the participant should rest. At the end of the 15 seconds, a bar plot appears displaying the root-mean-squared amplitude for each of the channels. You can repeat this step until all the channels have a low level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

'Start Visualization – 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 one grid, 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.



1. Visualize the 64 channels of your grid in real-time (buffer of five seconds)



2. Visualize the 64 channels of your grid and remove the noisy channels (enter for each column space-separated row numbers)

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

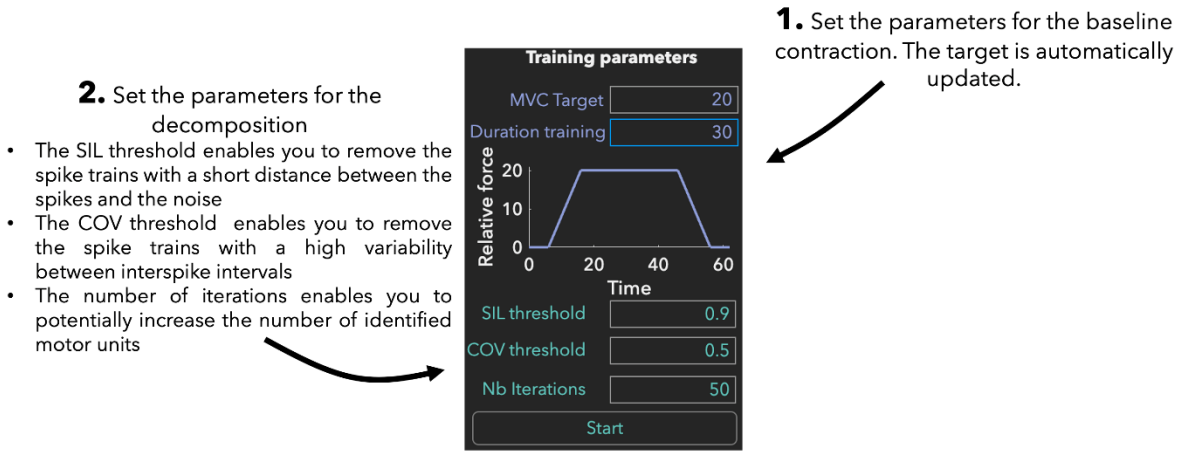
Training panel.

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

2. Check the box to remove the force offset from a small record

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.



The participant will then track a trapezoidal 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 optimize the decomposition (*ISpin_exported.m* or *ISpin.mlapp*). Specifically, you can change the number of channels generated by the signal extension function in the following line:

```
nbextchan = 1500;
```

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

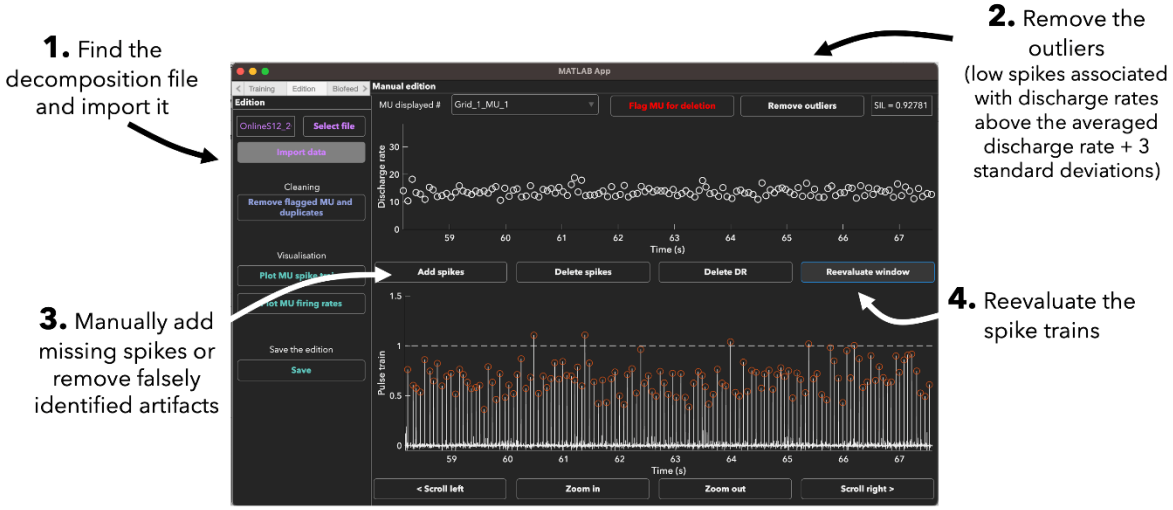
```
maxiter = 500; % max number of iterations for the fixed point algorithm
w = fixedpointalg(w, X, B , maxiter, 'logcosh');
```

The contrast functions 'square', 'skew', and 'logcosh' are available in *fixedpointalg.m*. We set the number of extended channels to 1500 and selected the contrast function 'logcosh' in the associated preprint.

Edition panel.

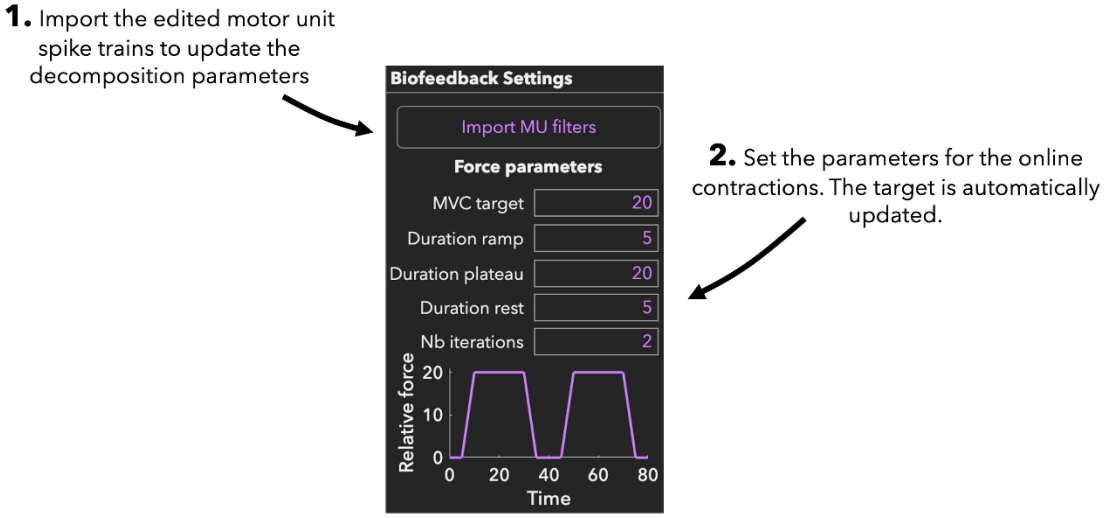
The manual edition step consists of visually checking the identified motor unit spike trains and manually correcting the automatic identification of spikes. The impact of manual edition on the decomposition can be observed in the associated preprint in Figure 3. For additional information about the full process to edit motor unit spike trains, please read the following paper that shows the high reliability of manual edition between operators and describes the successive steps of manual edition.

Hug F, Avrillon S, Del Vecchio A, Casolo A, Ibanez J, Nuccio S, Rossato J, Holobar A & Farina D. (2021). Analysis of motor unit spike trains estimated from high-density surface electromyography is highly reliable across operators. J Electromyogr Kinesiol 58, 102548.
https://www.sciencedirect.com/science/article/abs/pii/S1050641121000353?via%3Dihub

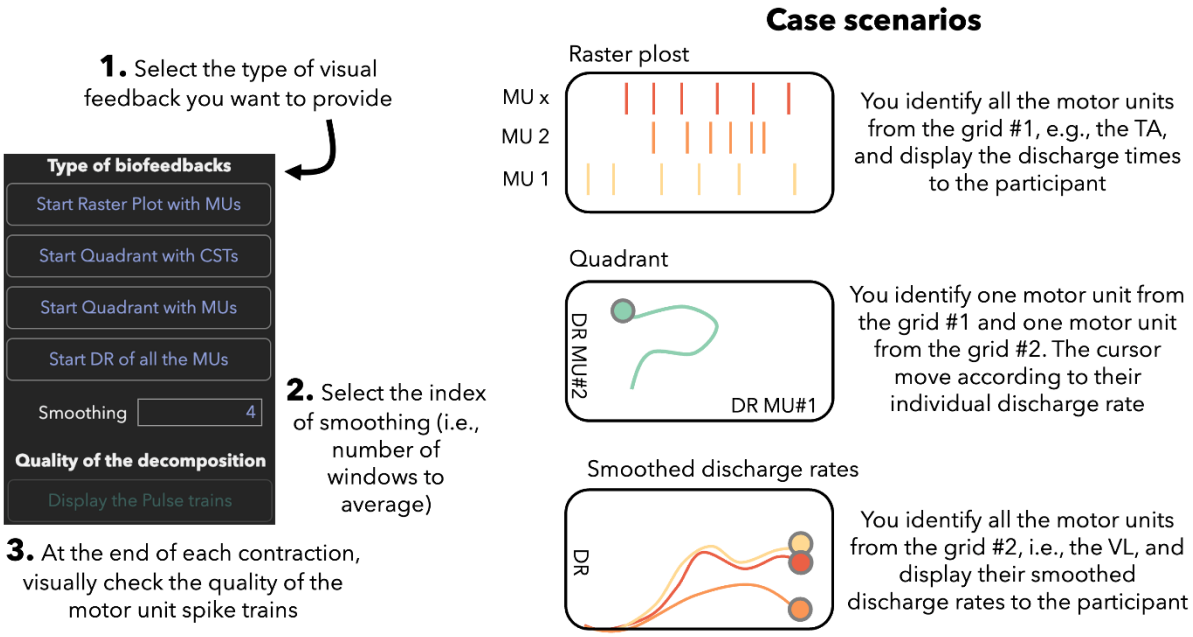


Once the edition of all the identified motor unit spike train is completed, i) click on the button 'Remove flagged MU and duplicates' to only keep the reliable and unique motor units and ii) click on 'Save' to save a file named 'Training…_edited' with the edited spike trains.

Biofeedback panel.



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 discharge activity of all the motor units, the decomposition is performed on the grid 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 grids displayed in the dropdown menus 'x axis…' and 'y axis…'. For the quadrant with a cursor moving according to the activity of two populations of motor units, the decomposition is performed on the grid displayed in the dropdown menus 'x axis…', and the motor units are separated in two groups to calculate x and y values.



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

Known bugs:

When pressing one of the buttons 'Start visualization', the following error message may appear in the Matlab command window and stops the process:

"Error using filtfilt>getCoeffsAndInitialConditions

Data length must be larger than 9 samples."

It is possible to re-click on 'Initialize SAGA' and re-try the button 'Start visualization' after waiting a few seconds.

Citation and technical support.

For technical assistance and support, please contact:
Dr. Julien ROSSATO
Laboratory of Neuromotor Physiology
IRCCS Fondazione Santa Lucia
Via Ardeatina 306
00179 Roma, Italy
E-mail address: julien.rossato14@gmail.com
ORCID: 0000-0003-4680-2933

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

Rossato, J., Hug, F., Tucker, K., Lacourpaille, L., Farina, D., Avrillon, S. (2023). I-Spin live: An open-source software based on blind-source separation for decoding the activity of spinal alpha motor neurons in real-time. eLife, 12:RP88670. https://doi.org/10.7554/eLife.88670.1