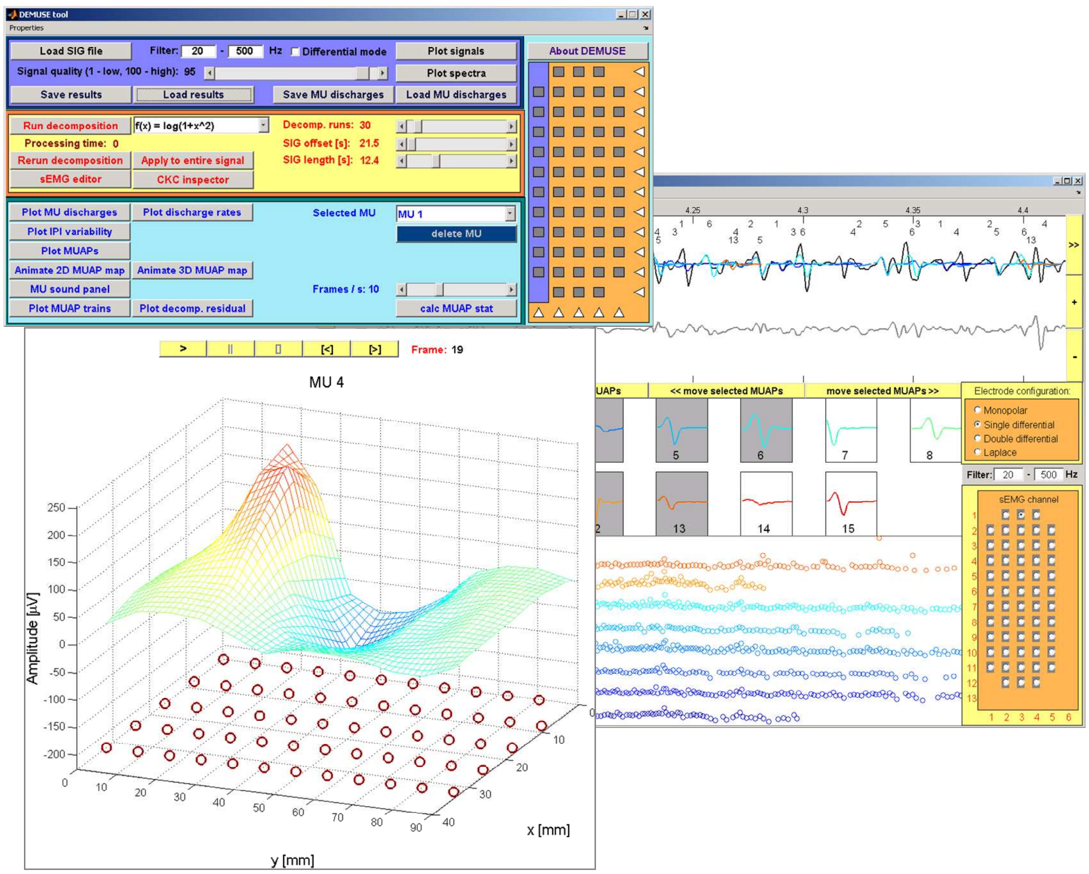

DEMUSEtool – user manual

Version 3.0

31/10/2008

This work was supported by a Marie Curie Intra-European Fellowship within the 6th European Community Framework Programme (DE MUSE, Contract No. 023537).

CONTENTS

1 Introduction

DEMUSEtool® is a matlab [1] program for visualization and decomposition of multichannel surface electromyograms (sEMG), acquired by EMG acquisition software v1.32 [2] or latter. It runs on a standard PC and enables the user to:

- load and visualize the multichannel surface electromyograms;
- decompose the sEMG signals into contributions of individual motor units (MUs);
- inspect and edit the results attained by automatic decomposition;
- display graphs of decomposition results, including plots of the MU discharge patterns, instantaneous discharge rate, motor unit action potentials (MUAPs) and their 2D and 3D animations;
- compare the original sEMG signals to the reconstructed MUAP trains;
- save and reload the decomposition results.

All the graphs are displayed as regular matlab figures and can be freely manipulated by standard matlab graphic tools (i.e., figure resizing, zooming, rotating, printing, etc.). User is referred to matlab documentation for further details on the use of matlab graphic user interface. For further information on EMG acquisition tool see [2].

Note: ! *The current version of the DEMUSEtool supports decomposition of isometric sEMG signals only, i.e., the signals, acquired during isometric muscle contraction. Intensive work on decomposition of dynamic sEMG signals is currently in progress and support for dynamic conditions will be built in the future versions of the DEMUSEtool*

2 DEMUSEtool components

DEMUSEtool represents the third layer in three-tier system architecture (**Figure 1**). The first two layers comprise the 128 channel EMG-USB electromyographic signal amplifier [4] and the sEMG acquisition software [2], respectively. sEMG signals under investigation are first acquired with a 2D matrix of electrodes. The matrix is put on the surface of the skin above the investigated muscle and connected to the EMG-USB signal amplifier [4], which amplifies and band-pass filters the signals and sends them to the sEMG acquisition software [2]. The acquisition software acquires the sEMG signals, displays them on the screen (for immediate visual inspection of the signal quality), and saves them into so called SIG files [2]. Information about the measurement session, including technical specifications (i.e., number of channels, sampling frequency, gain, etc.) is saved into purposely designed abstract file [2]. Finally, SIG and abstract files are loaded into DEMUSEtool and processed off-line (**Figure 1**).

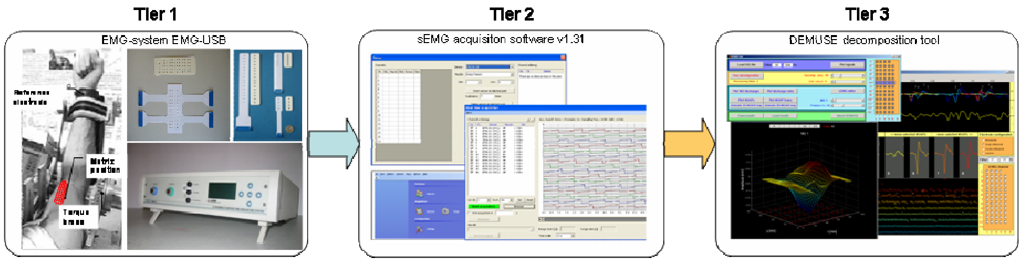

Figure 1: Three-tier architecture (with indicated data flows) of the sEMG decomposition system

3 DEMUSEtool installation

DEMUSEtool v3.0 is still a prototype and runs in the matlab programme environment [1]. As such, it is supported by several personal computer (PC) platforms, including Windows, Linux and Mac OS. It does not require any special hardware configuration. However, in the case of large number of channels and long sEMG signals, a sufficient amount of RAM (1 GB or more) should be installed on the system in order to prevent extensive swapping of the memory space.

3.1 Requirements

Minimal hardware configuration:

- 1 GHz CPU;
- 20 MB disk;
- 1 GB RAM.

Recommended hardware configuration:

- 2 GHz CPU or higher;
- 100 Mb disk;
- 2 MB L2 cache or more;
- 1 GB RAM.

To run DEMUSE tool following software should be properly installed:

- matlab [1], version 7.0 or higher.

3.2 DEMUSE files and folders

DEMUSEtool comprises several matlab's *.m and *.fig files which are located in the directory

`..\DEMUSEtool\programs\`

Program documentation is located in the directory

`..\DEMUSEtool\documentation\`

To install DEMUSEtool, copy both directories to your hard disk (e.g., to `c:\DEMUSEtool` directory) and set the path in matlab environment to `..\DEMUSEtool\programs\`

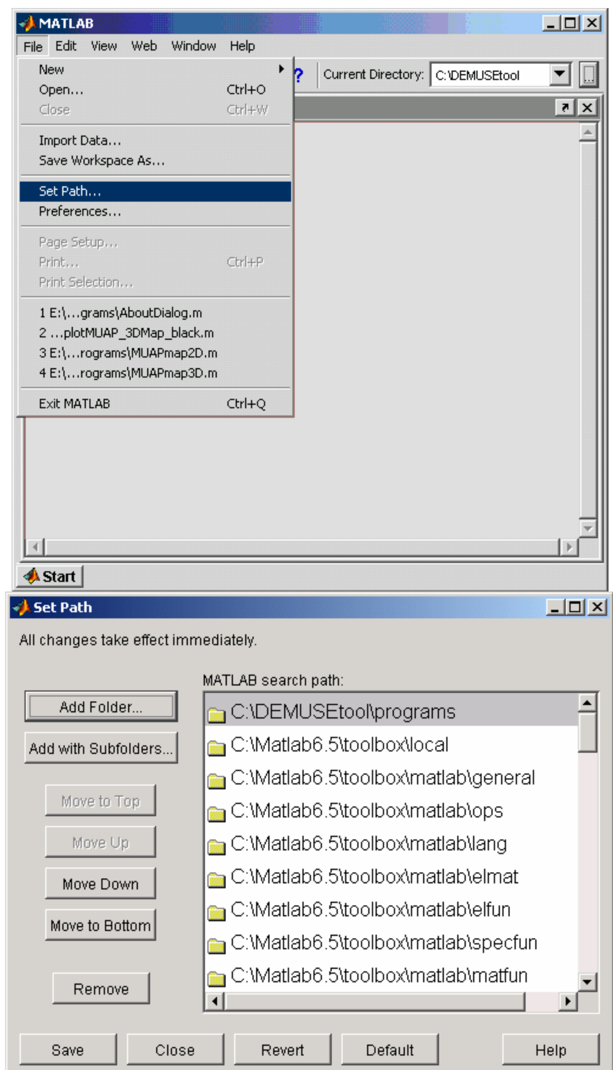

Figure 2: Setting the path in matlab programme environment (in this case, DEMUSE tool was copied to the directory `c:\DEMUSEtool`).

4 Using DEMUSE tool

4.1 Starting DEMUSE tool

To start the DEMUSEtool, type the following command to matlab command window:

```
>> DEMUSEtool
```

The main DEMUSEtool window appears (**Figure 3**). This window comprises four frames with the following groups of commands:

- loading, band-pass filtering and visualization of the acquired sEMG signals, saving and reloading of the decomposition results;
- decomposition of acquired sEMG signals and manual inspection of decomposition results;
- graphical plots and animations of the decomposition results;
- graphical schema of acquisition matrix configuration.

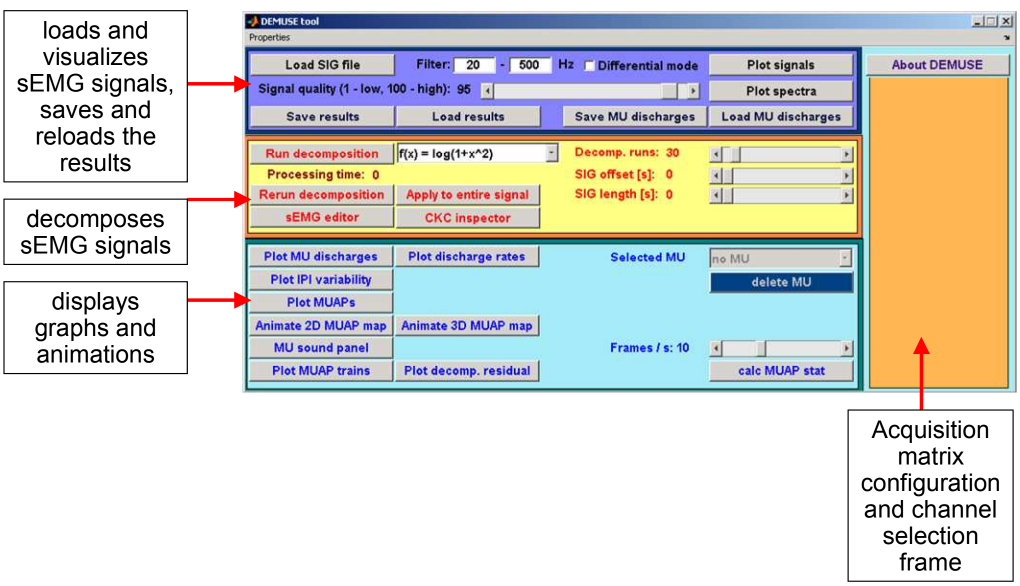

Figure 3: Main DEMUSEtool window with explanations of different command groups.

Each group of commands is depicted in different colour (**Figure 3**). Channel selection frame, displayed on the right-hand site in **Figure 3**, allows selection of different rows or columns of acquired sEMG channels. Its detailed description is provided in Subsections 4.2, 4.3 and 4.7.7.

4.2 sEMG signal loading

To load acquired sEMG into DEMUSEtool click on “`Load SIG file`” button (**Figure 4**, left panel). “`Load SIG file`” dialog window appears (**Figure 4**, right panel). Chose the SIG file and click on “`Open`” button.

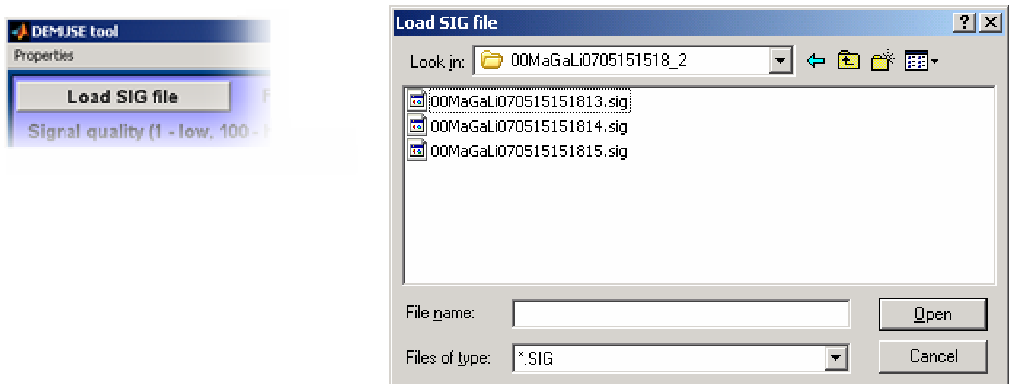

Figure 4: “`Load SIG file`” button (left panel) and “`Load SIG file`” dialog window (right panel).

sEMG acquisition software [2] supports arbitrary configurations and numbers of surface electrodes (up to 128 channels) and stores the information about the acquisition modalities into so called measurement session abstract file (XML). DEMUSEtool automatically loads this information (e.g., sampling frequency, dimensions of the acquisition system, electrode configurations, etc.) and displays the number of acquired channels and their relative spatial configuration in Channel selection frame (**Figure 5**). When requested information is missing, DEMUSEtool prompts for selection of a proper reader for sEMG files (**Figure 6**). There are several readers already implemented, supporting all main acquisition systems currently available in LISiN lab. For more complex acquisition configurations or other acquisition systems, specialized reader of the sEMG files can be implemented and added to the DEMUSE_readers directory (see Appendix I). DEMUSEtool will automatically load all the available readers and display their descriptions in the Reader Dialog Window (**Figure 6**). To select the specific reader, click on the line with its name. Corresponding reader description is automatically displayed in the Reader Description Panel. To confirm the reader selection and to continue with loading of the signal press OK button.

Proper reader can also be specified in a text file called DEMUSE_reader_Tag.txt (see Appendix I). Simply write the name of corresponding Matlab routine in DEMUSE_reader_Tag.txt file and copy the file into the directory with corresponding sEMG files. DEMUSEtool will automatically use the reader specified in DEMUSE_reader_Tag.txt for all the sEMG files in the corresponding directory.

In the case of loading failure, the Error Dialog Window will appear (**Figure 7**).

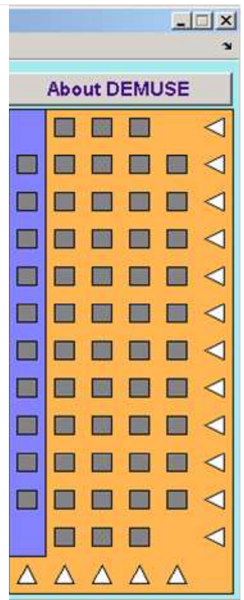

Figure 5: Channel selection frame displaying relative spatial organization of surface electrodes (grey rectangles). By clicking the white triangles, sEMG channels corresponding to specific row or column of surface electrodes can be selected for visual inspection. Selected column/row is denoted by blue rectangle.

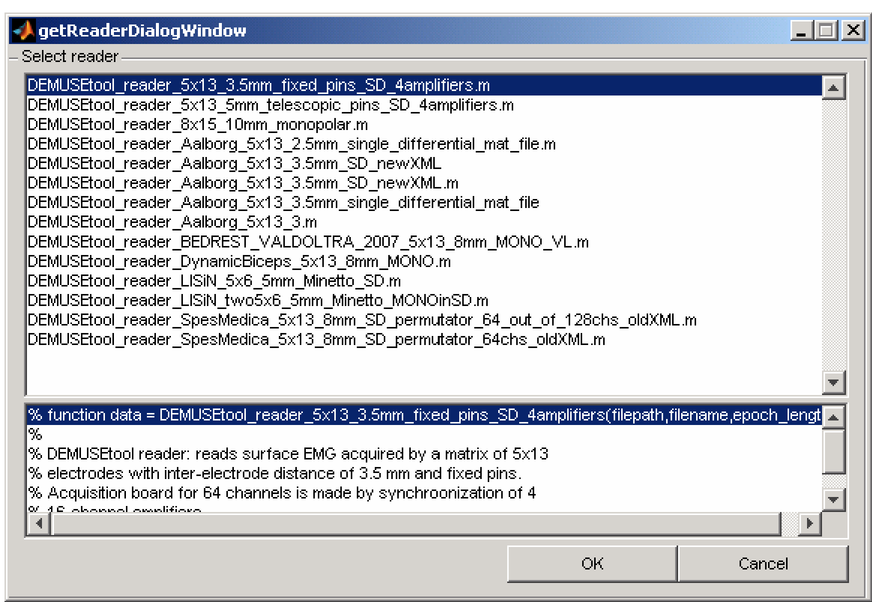

Figure 6: Reader Dialog Window for selection of the reader for sEMG signals. Upper panel displays all available readers, lower panel displays description of currently selected reader. Selection of the reader is confirmed by pressing the OK button.

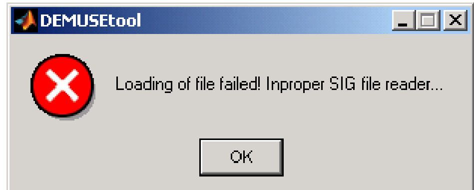

Figure 7: Error Dialog Window signalling the failure of sEMG loading.

DEMUSEtool uses 1st order Butterworth band-pass filter to filter the raw sEMG signals. Filter's cut-off frequencies can be controlled by typing new values into the text labels shown in **Figure 8**. Default cut-off frequencies are set to 20 and 500 Hz, respectively.

Figure 8: Text labels controlling the cut-off frequencies of built-in Butterworth band-pass filter.

When the number of MUs contributing to the sEMG signals is high, it might be beneficial to turn on the time differentiation of the sEMG channels (**Figure 9**). Time differentiator is a high-pass filter which suppresses the activity of small background MUAPs and enhances the discrimination of MUAPs from different MUs. Selection of time differentiator is optional and left to the user. A good practice is to play a bit with the time-differentiation and band-pass filtering before running the decomposition. The effect of time-differentiations and band-pass filtering can be examined by plotting sEMG channels and/or their power spectra (see Section 4.3).

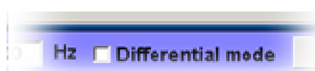

Figure 9: Check-box for selection of time differentiator. Time differentiator is a high-pass filter which suppresses the activity of small background MUAPs and enhances the differences between the MUAPs from different MUs. Differential mode should be selected in the case of high MU activity only.

DEMUSEtool automatically removes line interference and tests the acquired sEMG channels for presence of movement artefacts and bad skin-electrode contacts. However, percentage of the sEMG channels to be included into the decomposition must be specified explicitly by changing the value of signal quality slider (**Figure 10**). Setting the slider value to 95 %, for example, means that 5% of the channels with the lowest signal quality estimation will be skipped by the CKC decomposition. This selection does not influence the commands for plotting of sEMG channels and/or graphical representation of decomposition results (e.g. estimated motor unit action potentials). By default, all the available channels are included in all graphical presentations.

Figure 10: Slider for definition of sEMG signal quality. DEMUSEtool automatically removes line interference and tests the acquired sEMG channels for movement artefacts and bad skin-electrode contacts.

4.3 sEMG signal visualization

To display loaded, time-differentiated and/or band-pass filtered sEMG signals, select first offset and the length of signal interval to be displayed (**Figure 11**, left bottom panel), select the corresponding electrode row or electrode column (**Figure 11**, left panel) and click on "`Plot signals`" button (**Figure 11**, right panel). Matlab figure with selected sEMG channels appears (**Figure 12**). Zoomed-in version of **Figure 12** is depicted in **Figure 13**.

Note: ! *Due to the large number of acquired sEMG channels, only selected row/column of sEMG channels can be displayed in one figure. The number of figures, however, is not limited. You can display all the sEMG channels by consecutively selecting the different electrode columns, for example.*

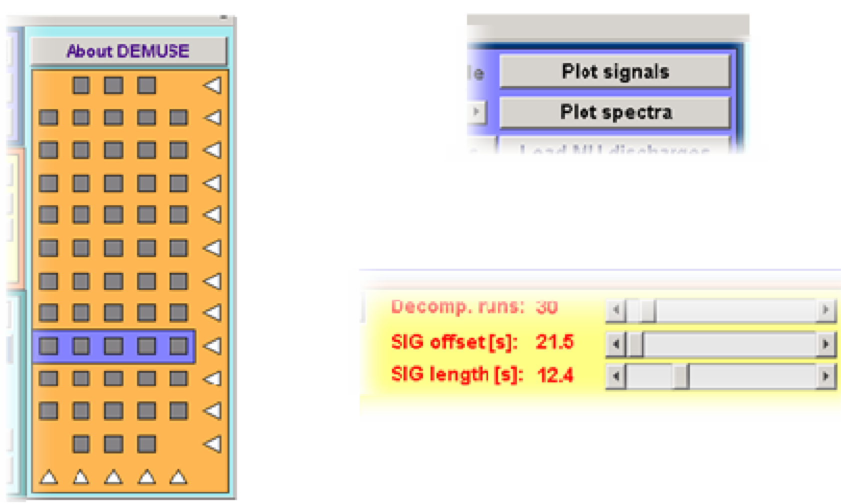

Figure 11: Channel selection frame with selected 9th row of electrodes (left panel) and plot signals, plot spectra buttons (right upper panel) and sliders for selection of offset and length of displayed signal interval (right bottom panel).

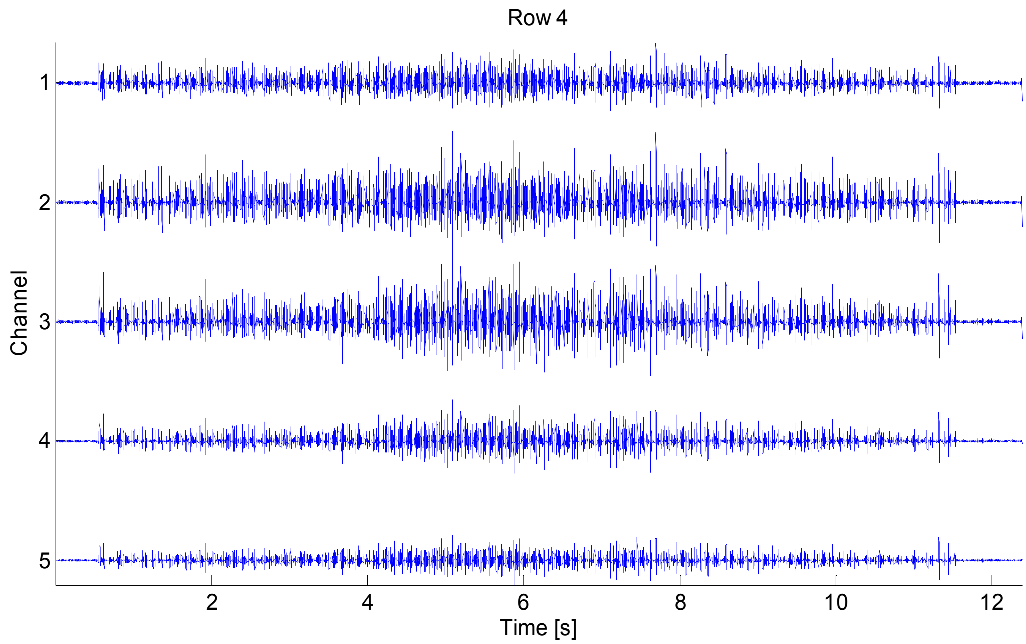

Figure 12: Matlab figure with selected sEMG channels

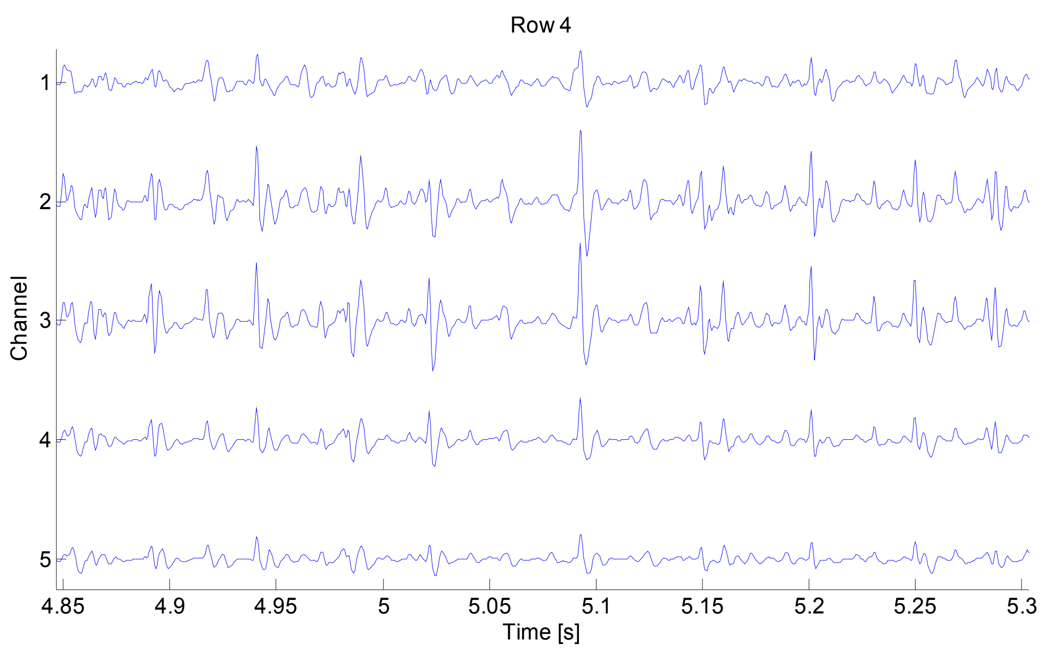

Figure 13: Zoomed-in version of **Figure 12**.

Displayed figures can be manipulated by using standard matlab graphical tools for zooming in/out, for saving and printing the figure (**Figure 14**). Figures can be closed by clicking on a corresponding buttons in the top right corner of each figure (**Figure 15**).

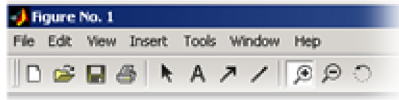

Figure 14: Matlab figure toolbar with tools for zooming in/out on a figure, and for saving and printing the figure.

Figure 15: Buttons for minimization, maximization and closing of the figure.

To display power spectra of band-pass filtered sEMG signals, click on "`Plot spectra`" button (**Figure 11**, right panel). Matlab figure with spectra of selected sEMG channels appears (**Figure 16**).

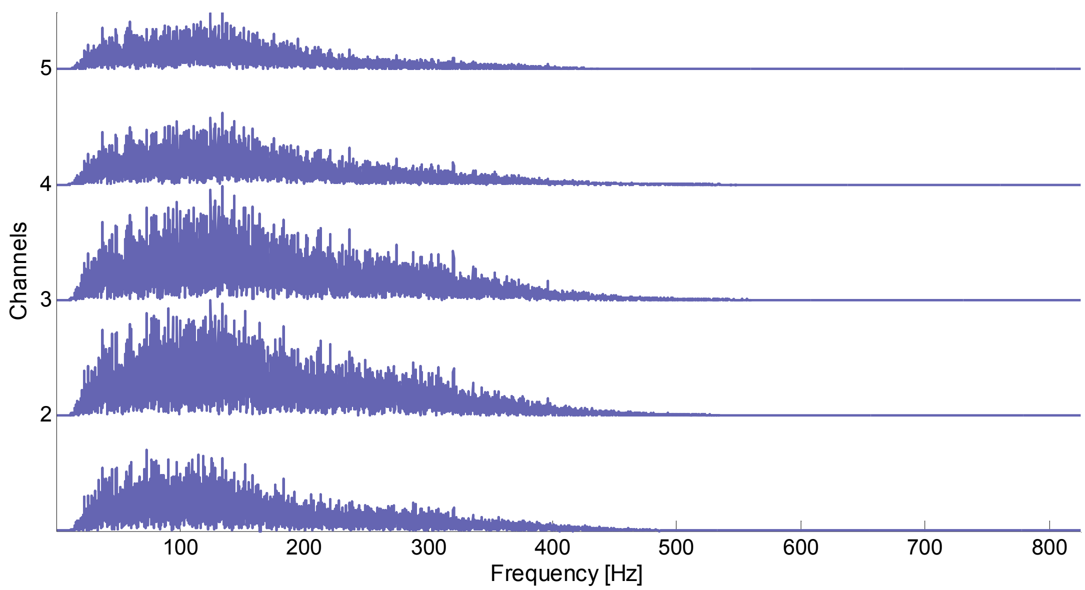

Figure 16: Matlab figure with power spectra of selected sEMG channels

4.4 sEMG signal decomposition

DEMUSEtool uses the gradient Convolution Kernel Compensation (gCKC) decomposition technique (see Appendix II and [3]). Decomposition is fully automatic, minimally biased by the properties of investigated muscle and nonparametric. The user specifies only the number of decomposition runs (**Figure 17**) and (optionally) nonlinearity used in gradient CKC optimization of estimated MU discharge patterns. User can select among the following possible values (see Appendix II for details about selection of gCKC nonlinearity):

Value of parameter	**Nonlinearity used by gradient CKC**
f(x) = log(1+x^2);	$\log(1+x^2)$
f(x) = log(1+x^4);	$\log(1+x^4)$
f(x) = x^2;	x^2
f(x) = 2*x/(1+x.^2);	$2x / (1+x^2)$
f(x) = x*log(1+x^2)-2*x+2*atan(x);	$x \cdot \log(1+x^2) - 2 \cdot x + 2 \cdot \mathrm{atan}(x)$

The initial offset and the length of sEMG time interval to be decomposed can also be selected by dragging the sliders in (**Figure 18**).

gCKC is a sequential motor unit (MU) identification technique and requires one iteration run per each reconstructed MU. The user can predefine maximal number of iterations by moving the slider "`Decomp. runs`" (**Figure 17**). As a general rule, the number of iterations should be larger or equal to the number of expected MUs (excluding the small and deep MUs, which contribute the background noise only). As the exact number of MUs is difficult to estimate, the number of decomposition runs should be large (default value is set to 30). DEMUSEtool automatically tests the reconstructed MU discharge patterns against the predefined ranges of physiological variables (i.e., discharge rate, variability of inter-pulse interval, etc.) and discards all the outliers. As a result, from 5 to 15 most reliably reconstructed MUs are taken into consideration. Reconstructed MUs are additionally sorted with respect to the aforementioned degree of decomposition reliability (the first MU being the most reliable one).

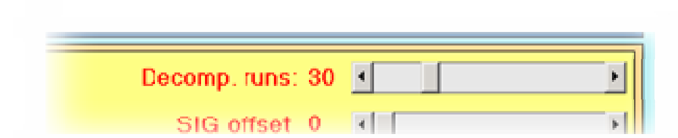

Figure 17: Slider for selection of decomposition runs

For practical reasons, the user can also specify the initial signal offset and length of sEMG interval entering into the decomposition (**Figure 18**). This allows discarding the initial signal portions where, for example, the contraction level is not yet stabilized, etc. The initial signal offset and interval lengths are measured in seconds.

Note: **!** *Due to the large number of sEMG channels and high memory consumption of gradient CKC method, the length of decomposition interval should generally be limited up to 20s. Longer signals should be divided into 20s long epoch which should be decomposed independently. The optimal length of decomposition interval depends also on the amount available computer memory.*

The decomposition starts by clicking on "`Run decomposition`" button (**Figure 19**).

Figure 18: Sliders for selection of initial offset and length of decomposition interval of sEMG signals.

Figure 19: "`Run decomposition`" button

During the decomposition, the reconstructed MU discharge patterns are displayed (**Figure 20**) and the instantaneous MU discharge rate and inter-pulse interval variability are automatically calculated. The reconstructed discharge pattern is put into the set of reconstructed MUs if and only if the calculated values fall within the expected range of values (i.e., average discharge rate between 5 and 60 pulses per second, Coefficient of Variability (CoV) of inter-pulse interval smaller than 50 %). The selected MU discharges of accepted discharge pattern are depicted by green circles (**Figure 21**). The reconstructed discharge pattern with the calculated values outside the expected range of values is discarded.

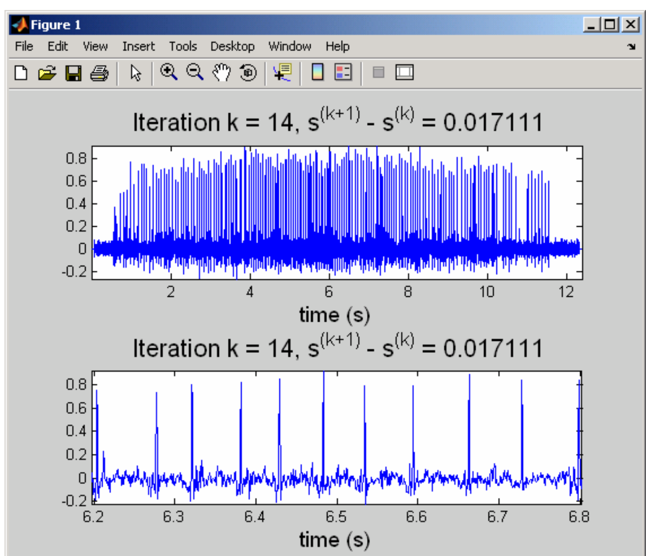

Figure 20: discharge pattern of individual MU, reconstructed in the 14th iteration of the gradient CKC decomposition technique. Each pulse (i.e., delta function) corresponds to a single MU discharge. Upper panel displays the entire train of reconstructed MU pulses (i.e., MU discharges), the lower panel displays its zoomed portion. Relative norm of difference between reconstructions of MU discharge pulses in two successive iteration ($s^{(k)}$ and $s^{(k+1)}$), divided by the norm of $s^{(k+1)}$, is used as a criterion to stop the iterations of gradient CKC.

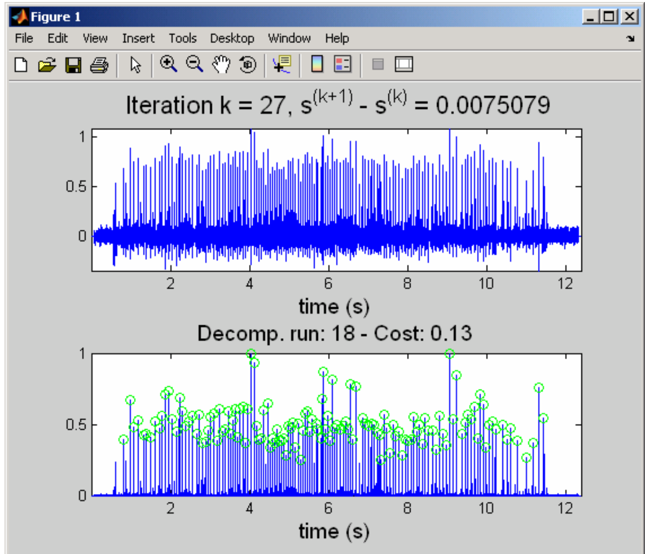

Figure 21: MU discharge pattern as reconstructed in the 27th iteration of the gradient CKC technique, after reaching the stop criterion $| s^{(k+1)} - s^{(k)} |/| s^{(k+1)}| < 0.01$. Each pulse corresponds to a single MU discharge. Average discharge rate and CoV of inter-pulse interval are within the expected range of values. Thus, in the lower panel, the selected MU discharges are depicted by green circles.

When the decomposition ends, the total processing time is displayed (**Figure 22**). We can now proceed to visualisation and saving of the results (Sections 4.6.1 and 4.8). Alternatively, we can change cut-off frequencies of band-pass filter, toggle time differentiation or select different nonlinearity and rerun the decomposition by clicking on "`Rerun decomposition`" button (**Figure 23**). Contrary to the "`Run decomposition`" button (**Figure 19**), "`Rerun decomposition`" button (**Figure 23**) keeps the record of already reconstructed MU discharge patterns and adds them to those reconstructed in the new decomposition run. "`Run decomposition`" button (**Figure 19**) automatically deletes previously reconstructed MU discharge patterns and starts the decomposition.

Figure 22: Text label displaying the total processing time.

Figure 23: "`Rerun decomposition`" button

Due to the large number of acquired sEMG channels and high memory consumption of gradient CKC method, the length of decomposition interval should generally be limited up to 20s. Longer signals should be divided into 20s long epoch which should be decomposed independently. Alternatively, the MU signatures in the space of discharge patterns can be reconstructed on a portion of a signal (i.e. by running the decomposition on the first 20s long epoch) and them applied to the entire signal length. This takes much less time than separate decompositions of different epochs, but is limited to MUs identified in the first time epoch only. MUs recruited at latter time moments (i.e., not active during the first epoch) will not be identified. In other words, if we decompose first out of four consecutive force ramps in **Figure 25**, for example, and then click on "`Apply to entire signal`" button, we will quickly retrieve entire discharge patterns of MUs identified during the first ramp (**Figure 25**), but will fail to identify MUs that were recruited after the first ramp, i.e. in second, third and/or fourth ramp only. A good practice is to decompose the signal epoch with highest expected number of active motor units (i.e., the last ramp in **Figure 25**), and then click on "`Apply to entire signal`" button

Figure 24: "`Apply to entire signal`" button

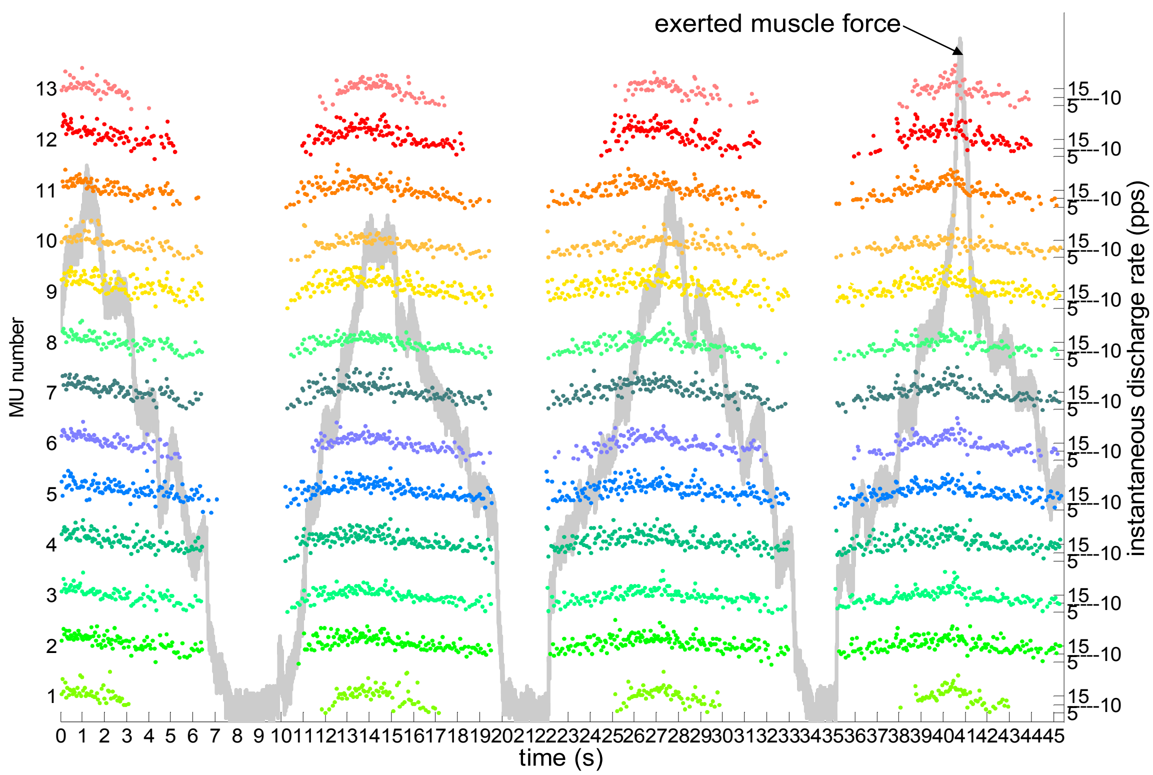

Figure 25: Result of "`Apply to entire signal`" button on an example of sEMG signals recorded during four force ramps of abductor pollicis brevis muscle. Only the first force ramp was decomposed by clicking on "`Run decomposition`" button, whereas discharge patterns on other three ramps were reconstructed by clicking on "`Apply to entire signal`" button. Each dot corresponds to a single MU discharge. Different MUs are denoted by different colours.

4.5 Editing of the decomposition results: sEMG Editor

DEMUSEtool integrates two toolkits for visualization and inspection of decomposition results. First one, so called `sEMG editor`, allows editing of single channels and mimics the user-interface of EMGLAB – a popular intra muscular decomposition tool [7]. User can easily switch among different sEMG channels, but only one channel is display at a time.

The second toolkit, so called `CKC inspector`, allows inspecting the raw outputs of CKC method, i.e., trains of delta pulses, and displays the multichannel MUAPs as detected by the entire acquisition matrix. In the sequel, features of sEMG editor are briefly explained, whereas CKC inspector is discussed in Section 4.6.

To open sEMG editor window (**Figure 27**), click on “`sEMG editor`” button (**Figure 26**).

Figure 26: “`sEMG editor`” button

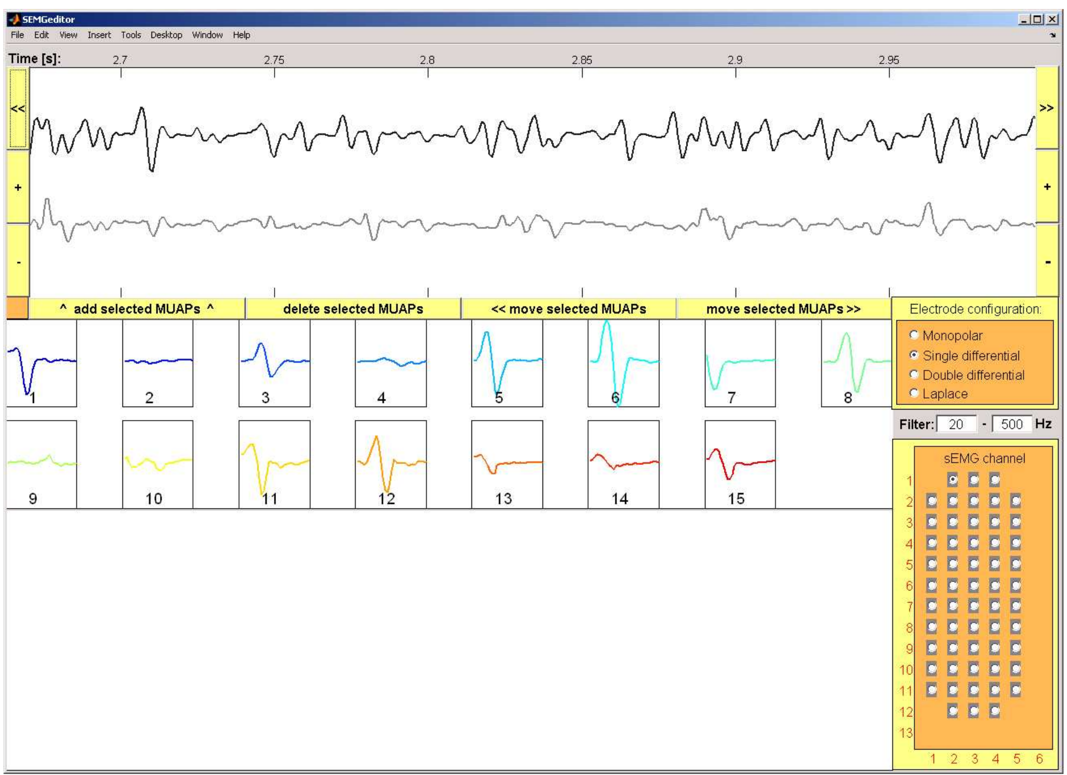

Figure 27: “`sEMG editor`” window. For details, see Subsections 4.5.1 – 4.5.5.

4.5.1 Plotting sEMG signals

`sEMG editor` window comprises three panels (**Figure 27**). The top panel displays original sEMG channel (black line) and the residual after subtraction of identified MUAPs (gray line). The ID of MU discharging at a particular time moment is also depicted (**Figure 28**).

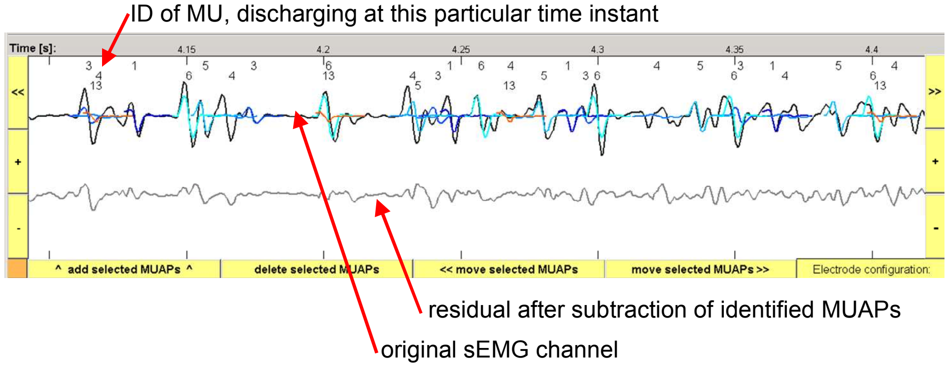

Figure 28: The top panel of "`sEMG editor`" window with original sEMG channel (black line) and residual after subtraction of the identified MUAPs (gray line). MUAP templates of identified MUs are superimposed over the original sEMG channel, their IDs are listed on the top of each template.

Buttons at the left and right hand side of the top panel determine the position and the size of the time window used for depicting the sEMG channel. Buttons "<<" and ">>" move the window left and right with respect to the sEMG channel, respectively (**Figure 29**). Button "+" ("-") zooms in(out) on a displayed signals (decreases/increases the length of the time window).

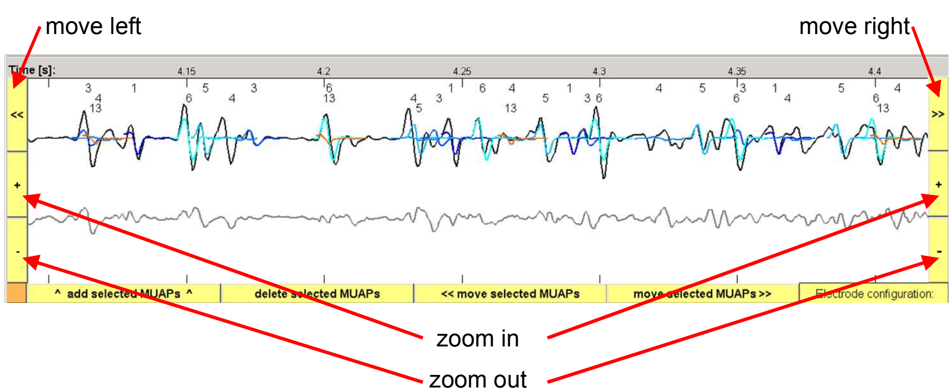

Figure 29: The top panel of "`sEMG editor`" window with buttons for moving left/right and zooming in/out on displayed signals.

4.5.2 Displaying instantaneous MU discharge rate

The central panel of the "`sEMG editor`" window displays the MUAP templates of different MUs, as estimated by spike triggering averaging of the selected sEMG channel (**Figure 30**). The identified MU discharges are taken as triggers. The length of the averaging window is set equal to 25 ms.

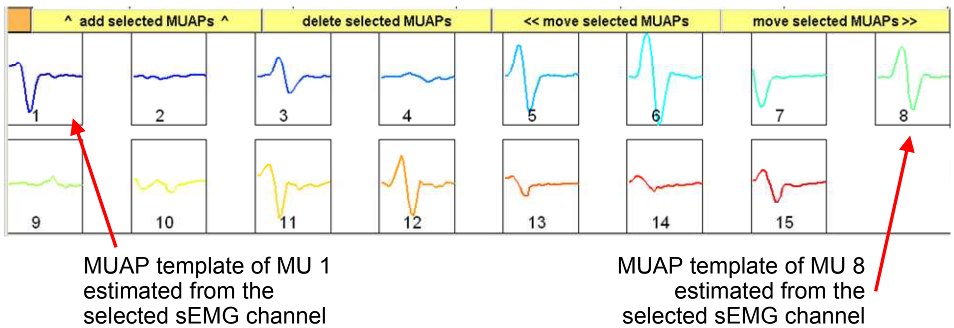

Figure 30: The central panel of "`sEMG editor`" with MUAP templates estimated by spike triggered averaging of the selected sEMG channel.

The central panel allows the user to select a particular MU by clicking on corresponding MUAP template. The MUAP occurrences of selected MUs are automatically displayed in the top panel, while the bottom panel displays the instantaneous discharge rate plot (**Figure 31**). Several MUs can be simultaneously selected (**Figure 32**).

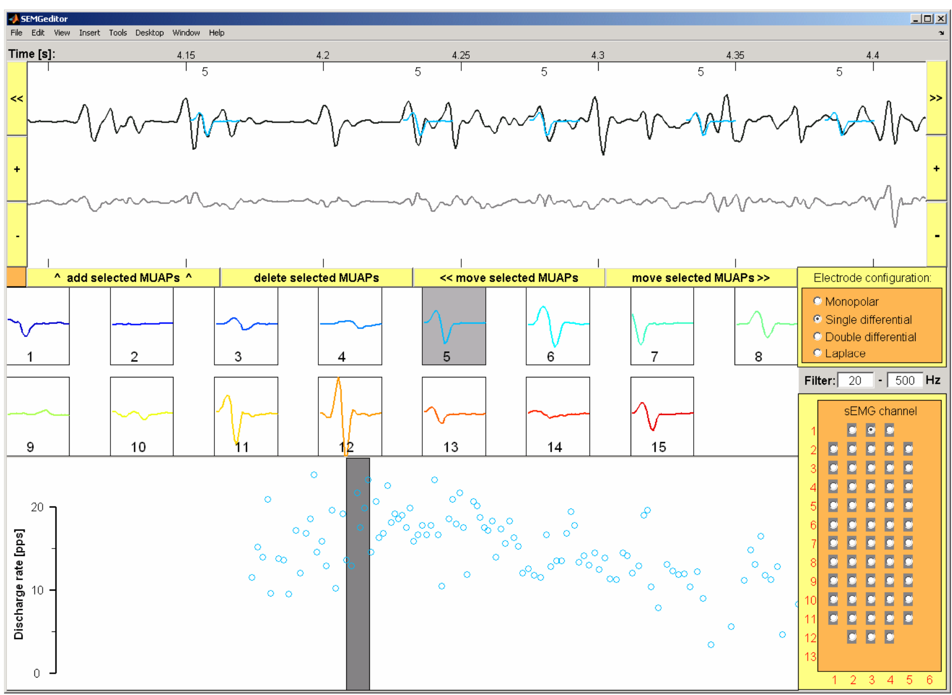

Figure 31: MU 5 is selected by clicking on its MUAP template in the central panel. MUAP occurrences and instantaneous discharge rates of selected MU are depicted in the top and the bottom panel, respectively.

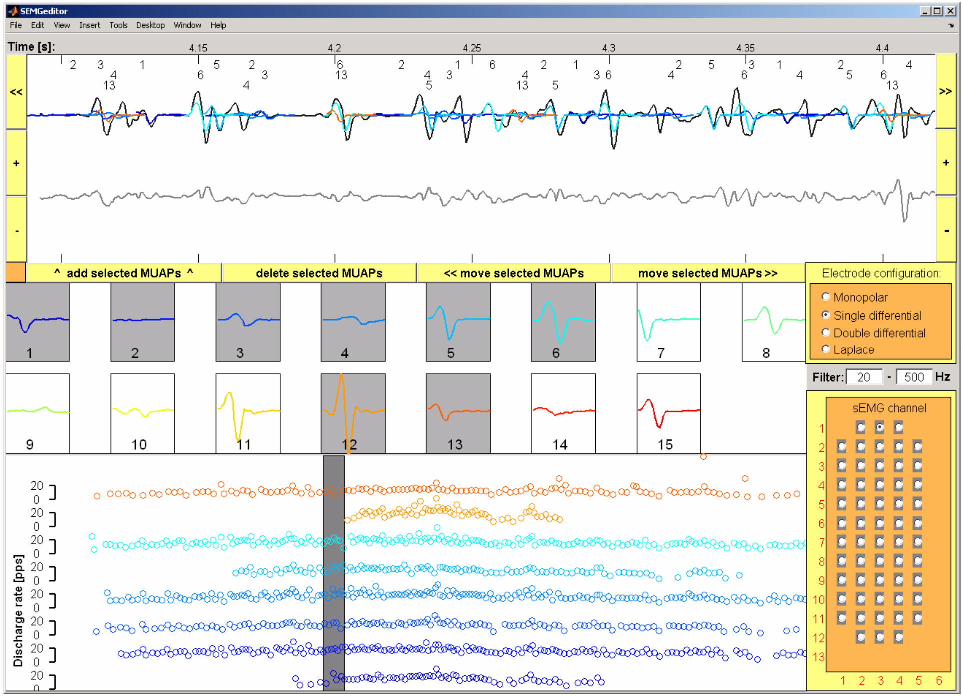

Figure 32: The same as in **Figure 31**, with all MUs selected.

Each circle in the bottom panel corresponds to a single MU discharge (**Figure 33**). The horizontal position of the circle denotes the MUAP occurrence time, whereas its vertical position reflects instantaneous MU discharge rate (calculated as the quotient between the sampling frequency and the inter-pulse interval preceding the selected MU discharge). The position and the length of the signal window, displayed in the top panel, is also depicted (grey rectangle). User can move to the time of a particular MU discharge by simply clicking on the corresponding circle.

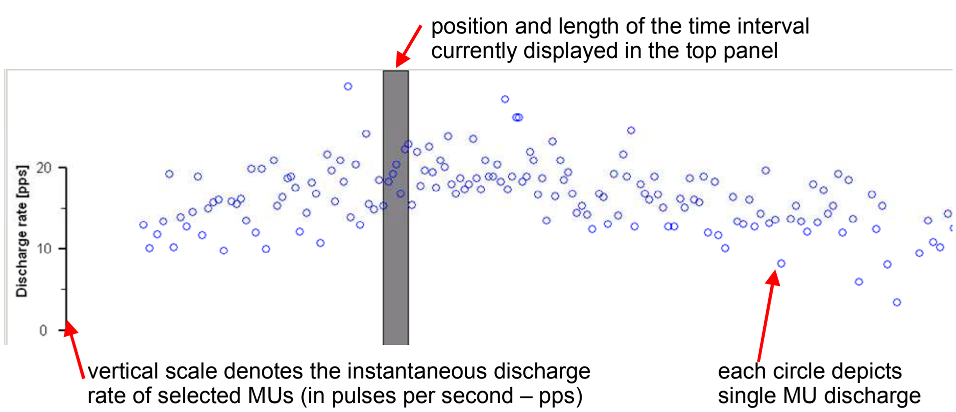

Figure 33: Bottom panel of the "`sEMG editor`" window with the instantaneous discharge rate plot of a selected MUs.

4.5.3 Adding a new MUAP occurrence

A double click on a MUAP template in the central panel, followed by a click on the "`add selected MUAP`" button adds the MUAP of the selected MU to the top panel. The optimal MUAP position is automatically determined by the minimum squared error between the residual signal (**Figure 34**, gray line) and the selected MUAP template. In the central panel, the selected MUAP template is denoted by a red rectangle (**Figure 34**, central panel). A further double-click on the already selected MUAP template (e.g., MUAP template of MU 12 in **Figure 34**) deselects the corresponding MU.

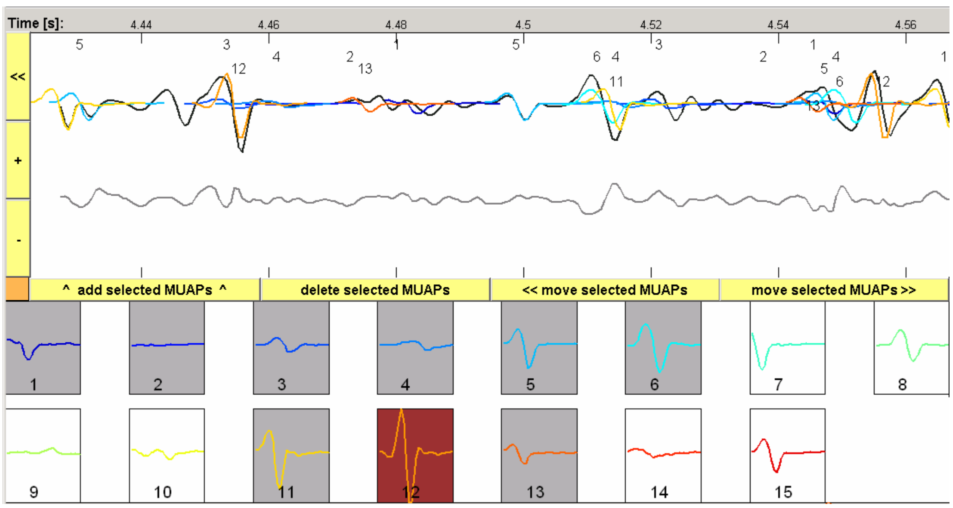

Figure 34: Double click on a MUAP template of MU 12, followed by a click on a "`add selected MUAP`" button adds the MUAP to the top panel. The MUAP template selected by double-click is depicted by a red rectangle.

Note: ! *Double-click on a MUAP template could directly result in adding of the new MUAP occurrence to the top panel. However, this would cause errors in the case of unintentional double-clicks on the central panel. Therefore, adding of a MUAP occurrence must be explicitly confirmed by clicking on* "`add selected MUAP`" *button.*

4.5.4 Moving and deleting MUAP occurrences

Each MUAP occurrence, displayed in the top panel, can be manually moved to the left, moved to the right, and deleted, respectively. The user must first click on the displayed MUAP occurrence in the top panel in order to select it. The selected MUAP occurrence is denoted by a thick red line (**Figure 35**). Afterwards, the selected MUAP occurrence can be moved or deleted by clicking on a "`<< move selected MUAPs`", "`move selected MUAPs >>`" and "`delete selected MUAPs`" button, respectively).

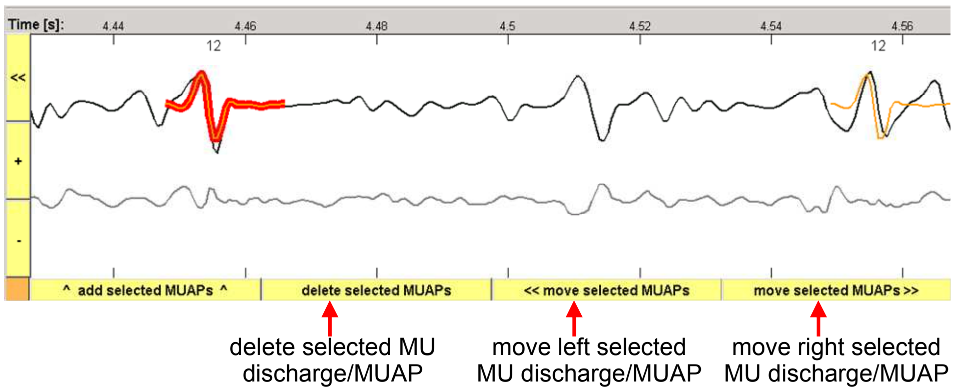

Figure 35: Manipulation of selected MUAP occurrence (depicted by red thick line). The selected MUAP occurrence can be moved left/right or deleted

Each user action results in an immediate update of a signal residual and MUAP templates (**Figure 36**). Selected MUAP occurrence can be deselected by a single click.

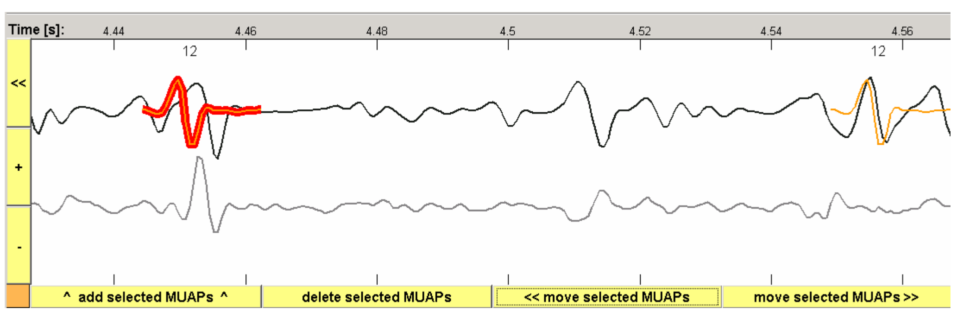

Figure 36: Same as in **Figure 35**, with the selected MUAP occurrence moved to the left. Increase of the sEMG channel residual signal (grey line) is clearly visible (when compared to **Figure 35**).

Note: *Several MUAP occurrences can be simultaneously selected. This saves the user's time and energy in the case of intensive manual editing.*

4.5.5 Selecting sEMG channel, spatial filter and cut-off frequencies

The sEMG channel to be displayed can be selected by clicking on a radio button representing the corresponding channel in the left bottom corner of the "`sEMG editor`" window (**Figure 37**). Electrode rows and columns are denoted by red numbers displayed at left/bottom of the channel selection panel. Text labels right above the "`sEMG channel`" selection panel, display cut-off frequencies of the 1st order Butterworth band-pass filter. Cut-off frequencies can be modified by typing into the aforementioned text labels (**Figure 37**). The sEMG signals and corresponding MUAP templates are automatically recalculated when cut-off frequencies are changed. Default cut-off frequencies values are set to 20 and 500 Hz, respectively.

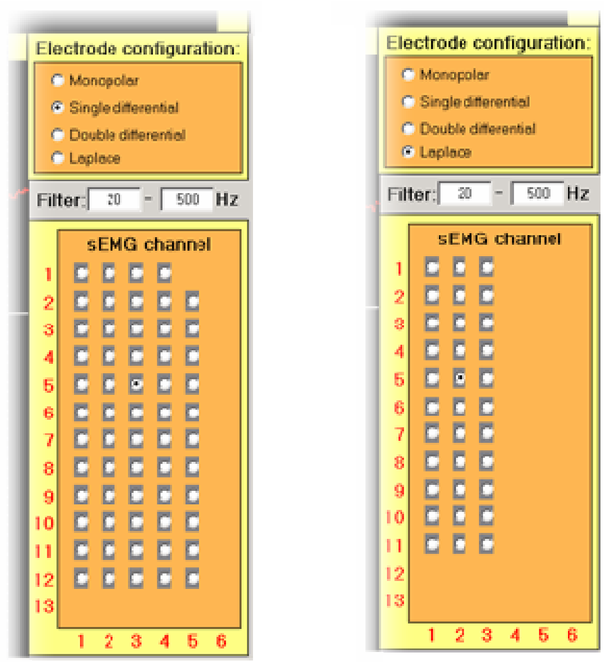

Figure 37: Command frame of "`sEMG editor`" with "`sEMG channel`" panel, Butterworth band-pass filter cut-off frequencies and "`Electrode configuration`" panel for single differential electrode configuration (*right*) and Laplacian electrode configuration (*left*).

Above the filter cut-off frequencies, there is the "`Electrode configuration`" panel (**Figure 37**). It allows the user to choose among different spatial filters applied to acquired sEMG signals. The following four spatial filters are currently supported: monopolar, longitudinal single differential (default), longitudinal double differential and Laplacian. "`sEMG channel`" selection panel, sEMG channel displayed in the top panel, and corresponding MUAP templates are automatically updated whenever a new spatial filter is selected (**Figure 37**).

4.6 Editing of the decomposition results: CKC inspector

CKC inspector allows editing the raw outputs of the gradient CKC method, i.e., trains of delta pulses. Inspector is launched by first selecting the MU from the "`Selected MU`" pup-up menu (**Figure 38**) and then clicking on the "`CKC inspector`" button (**Figure 39**).

Figure 38: "`Select MU`" pup-up menu allows selection of MU to be edited by CKC inspector.

Figure 39: "`CKC inspector`" button.

CKC inspector window consist of two panels (**Figure 40**). Lower panel displays discharge pattern of selected MU, as estimated by gradient CKC method. Two different versions of the same discharge pattern are depicted: train of delta pulses as calculated by gradient CKC method (upper part of the panel in **Figure 41**) and instantaneous discharge rate of estimated MU (lower part of the panel in **Figure 41**). User can zoom-in and zoom-out on time axis by clicking on buttons 'II' and 'I I', respectively (**Figure 42**). By clicking buttons '<' and '>' (**Figure 42**) the displayed portion of the MU discharge pattern is moved left and right, respectively.

MU discharges can be added or deleted by clicking the '`add discharge`' and '`delete discharge`' buttons. After each click on these buttons, mouse pointer changes from arrow to full cross. Drag the cross to the pulse to be added/deleted and left-click to add/delete the MU discharge (**Figure 43**). Several MU discharges can be deleted simultaneously by clicking on '`delete many discharges`' button (**Figure 42**). In this case, full-cross pointer is used to determine the left and the right edge of the MU discharge cancellation interval. All the discharges between the aforementioned edges will be deleted.

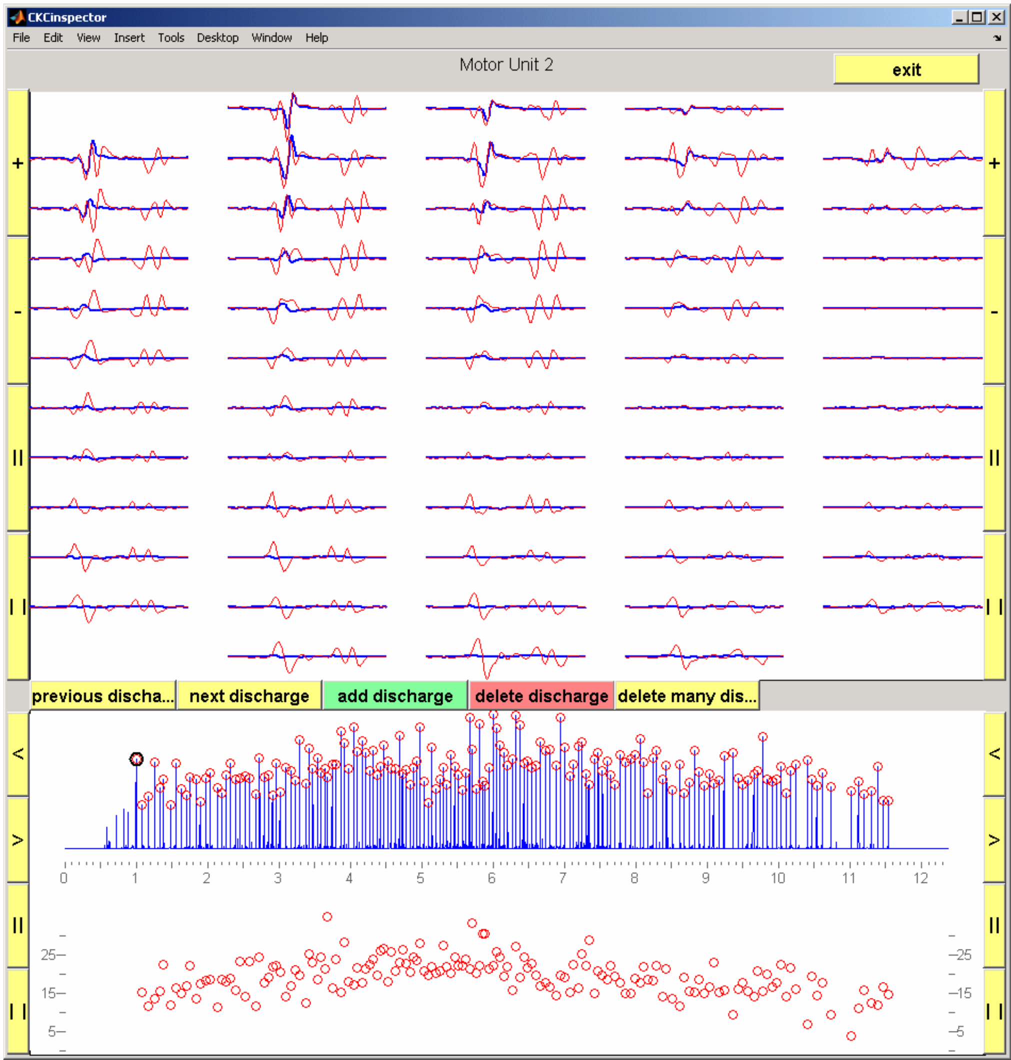

Figure 40: Window of "`CKC inspector`" with upper panel displaying multichannel MUAP of selected MU (as detected by all surface electrodes) and lower panel displaying the train of MU discharge times as estimated by CKC decomposition method.

Upper panel of the "`CKC inspector`" window displays multichannel MUAP template of selected MU as detected by all surface electrodes and estimated by spike triggered averaging of sEMG channels (blue thick lines in **Figure 40**). All available MU discharges are used as triggers. Displayed MUAPs are spatially organized in rows and columns, reflecting the relative position of pick-up electrodes. By right-clicking on a red circle in the lower panel (**Figure 41**), portions of the original sEMG channels around that discharge are displayed in the upper panel of "`CKC inspector`" window (red thin lines in **Figure 40**) and aligned with the displayed MUAP templates. Selected MU discharge is depicted by black thick circle in the lower panel

(**Figure 40**). This allows inspection of MUAP presence and MUAP superimpositions on all the sEMG channels simultaneously. By clicking on '`previous discharge`' ('`next discharge`') button (**Figure 42**), sEMG portion around the previous (next) MU discharge is displayed. The use of buttons, which control the length and the scale of displayed MUAP template and portions of sEMG is explained in **Figure 44**.

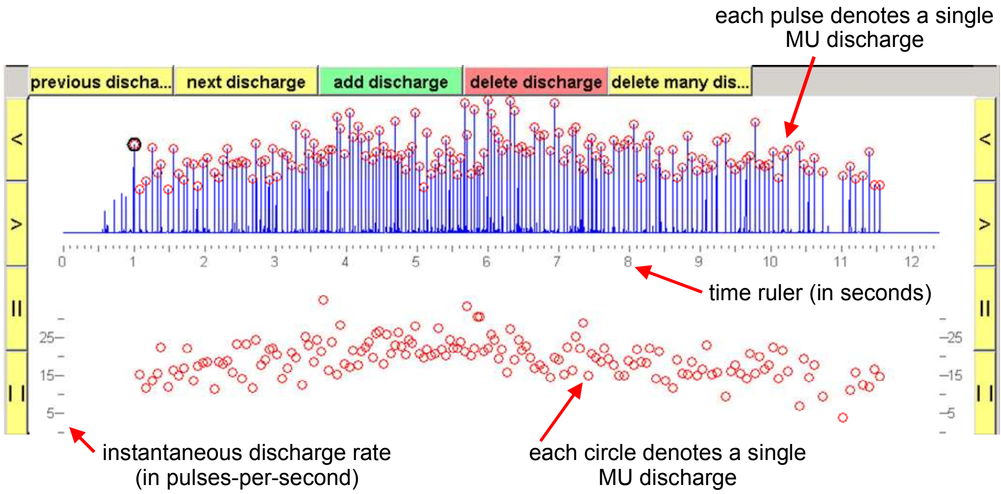

Figure 41: Lower panel of "`CKC inspector`" window with train of delta pulses as estimated by CKC method in the upper part of the panel, and instantaneous discharge rate plot in the lower part of the panel. In both plots, horizontal ruler denotes the time (in seconds). Vertical axis of the lower plot denotes the instantaneous discharge rate (in pulses-per-second).

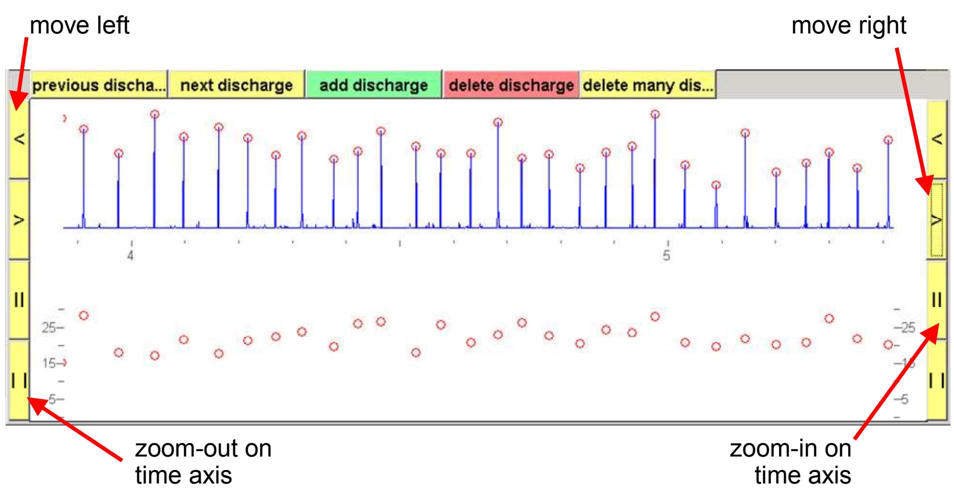

Figure 42: Same as in **Figure 41** with the time axis zoomed-in. Delta pulses denoting the discharge times of single MU are clearly visible. Base-line noise is negligible and inter-discharge interval exhibits regular behaviour. This gives us confidence in the results of CKC decomposition. Buttons on the edges of the lower panel control the size and position of the displayed portion of the MU discharge pattern.

Figure 43: Portion of a MU discharge pattern (same as in **Figure 42**) before (*left panel*) and after cancellation of the central MU discharge (*right panel*). In the right panel, full-cross pointer used for selection of the MU discharge is partially visible. MU discharge is deleted by clicking on the '`delete discharge`' button, positioning the full-cross pointer over the MU discharge (i.e. pulse) and clicking the left mouse button.

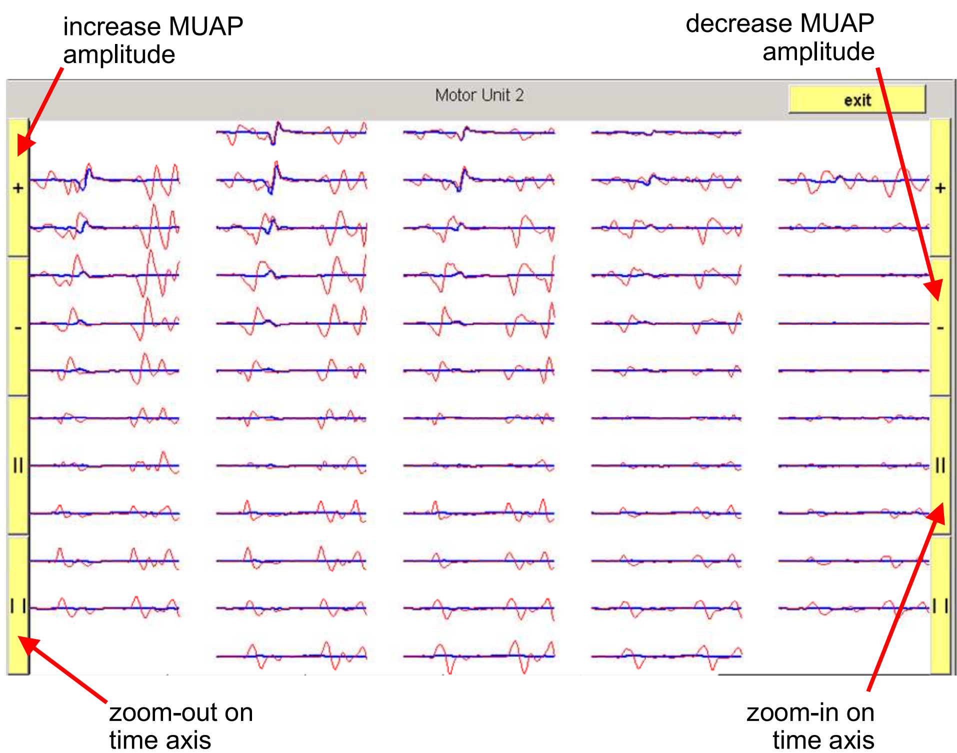

Figure 44: Buttons for controlling the length and the scale of the displayed multichannel MUAP template and raw sEMG portions. Buttons '+' and '–' scale the MUAP/sEMG amplitude. Buttons 'II' and 'I I' zoom in and out on the time axis, respectively.

4.6.1 Deleting of decomposition results

Reconstructed discharge patterns of specific MU can be deleted by first selecting the MU in a “`Selected MU`” pup-up menu and then clicking on a “`delete MU`” button (**Figure 45**). A window opens for conformation of MU cancellation (**Figure 46**). To really delete the MU, click on ‘`Yes`’ button. To return to DEMUSEtool without deleting the MU, click on ‘`No`’ or ‘`Cancel`’.

Figure 45: “`delete MU`” button (*left panel*) and “`Selected MU`” pup-up menu (*right panel*).

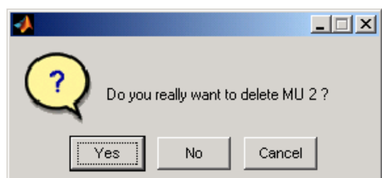

Figure 46: window for conformation of MU cancellation.

4.7 Graphical results

DEMUSEtool includes several tools for graphical representation of the decomposition results. The user can plot the discharge patterns of reconstructed MUs, instantaneous and smoothed MU discharge rates, multichannel MUAPs and reconstructed MUAP trains. In addition, MUAP generation, propagation and attenuation can be animated for each identified MUs. All the graphical results are depicted in Matlab figures and can be easily manipulated by standard Matlab's editing tools. Background colour of all the plots can be selected in "`Properties`" menu (**Figure 47**). In the sequel, description of each aforementioned graphical representation is provided.

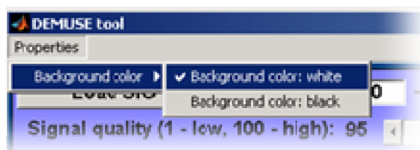

Figure 47: "`Properties`" menu allows selection of a background colour of all the plots, of CKC inspector and of sEMG editor.

4.7.1 MU discharge patterns plot

MU discharge patterns are plotted by a click on a "`Plot MU discharges`" button (**Figure 48**). A matlab figure opens with a plot of all reconstructed MU discharge patterns (**Figure 49**). Each circle in the figure corresponds to a single MU discharge. The horizontal position of the circle denotes the time of MU discharge, whereas its vertical position reflects instantaneous MU discharge rate (calculated as a quotient between the sampling frequency and the inter-pulse interval preceding the given MU discharge). Discharge patterns of different MU are depicted one above the other.

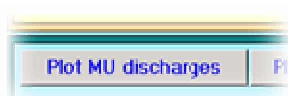

Figure 48: "`Plot MU discharge`" button

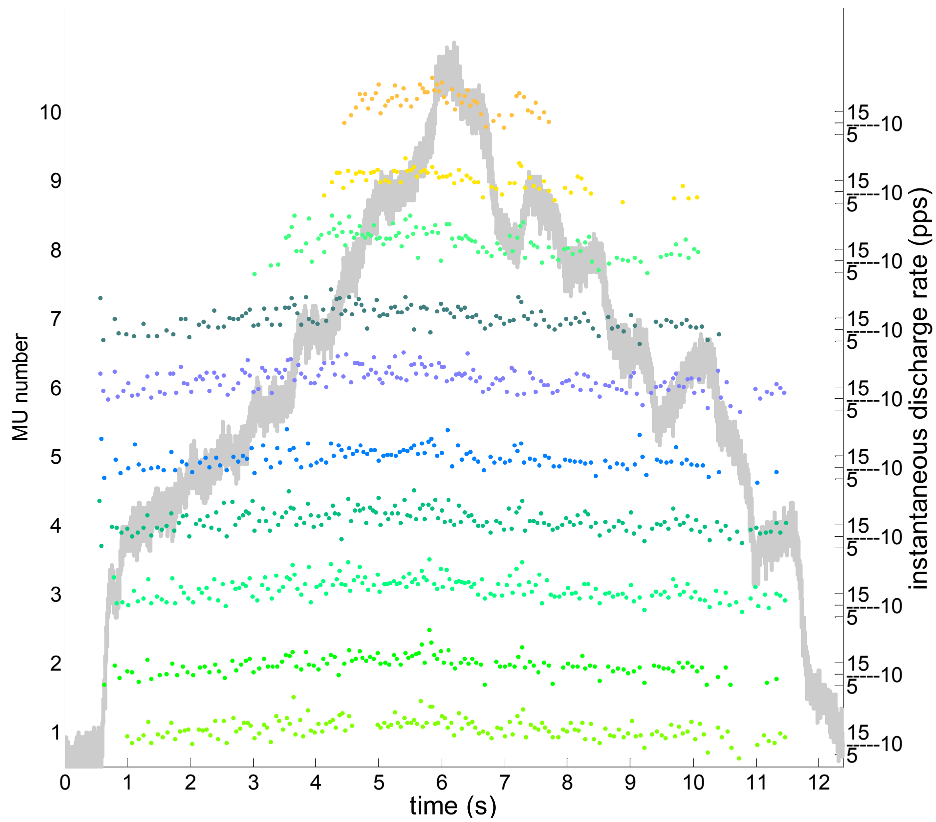

Figure 49: Plots of reconstructed MU discharge patterns. Vertical axis on the left displays MU IDs, vertical axis on the right displays the instantaneous discharge rates (in pulses per second - pps). The tick lines on the right denote the discharge rates of 5, 10 and 15 pps, respectively. Discharge patterns of different MUs are depicted one above the other.

4.7.2 Smoothed discharge rate plots

Smoothed MU discharge rates are plotted by a click on a "`Plot discharge rates`" button (**Figure 50**). A matlab figure opens (**Figure 51**) with a different colour lines depicting the smoothed discharge rates of different MUs (one line per each MU). The thick grey line depicts the exerted muscle force (when measured during the acquisition of sEMG signals). Smoothed discharge rates are calculated by low-pass filtering of the instananeous discharge rates (1st order Butterworth filter with cut-off frequency set to 3 Hz).

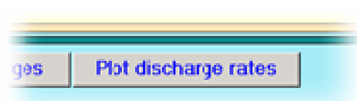

Figure 50: "`Plot discharge rates`" button

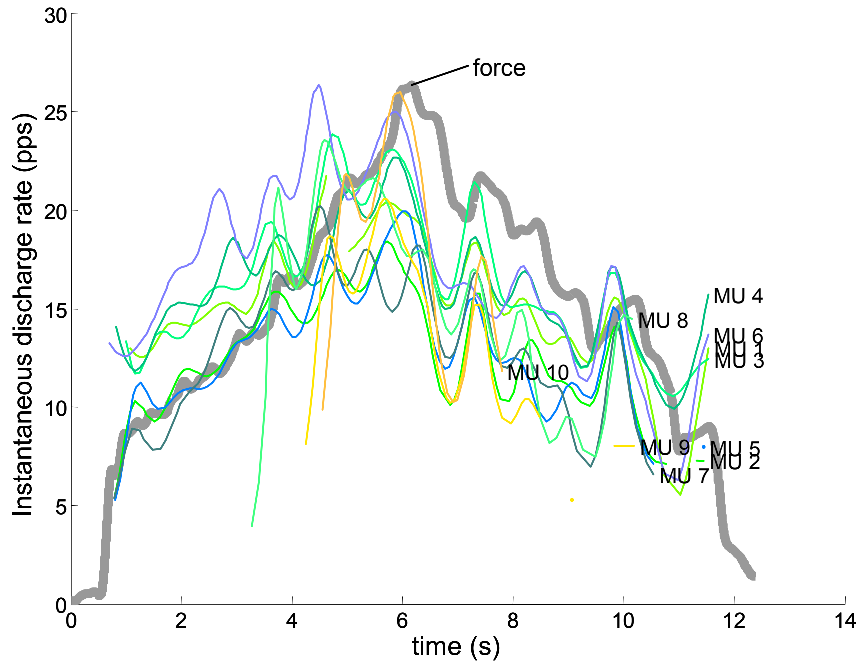

Figure 51: Plot of instantaneous discharge rates (coloured thin lines). Vertical axis depicts the instantaneous discharge rates (in pulses per second - pps). Thick grey line depicts the measured muscle force.

4.7.3 Inter-pulse variability

Variability of inter-pulse interval (IPI) of reconstructed MU discharge patterns is displayed by selecting the MU in a "`Selected MU`" pup-up menu and clicking on a "`Plot IPI variability`" button (**Figure 52**). A window opens with four different plots (**Figure 53**): inter-pulse interval (*upper left panel*), smoothed discharge rate (*upper right panel*), Coefficient of Variability (CoV) of IPI (*bottom left panel*) and CoV of IPI versus IPI mean (*bottom right panel*). Smoothed discharge rate is calculated by low-pass filtering the instantaneous discharge rate with 1st order Butterworth filter (cut-off frequency of 3 Hz). Standard deviation of IPI is calculated over 10 consecutive MU discharges.

Figure 52: "`Plot IPI variability`" button (*left panel*) and "`Selected MU`" pup-up menu (*right panel*).

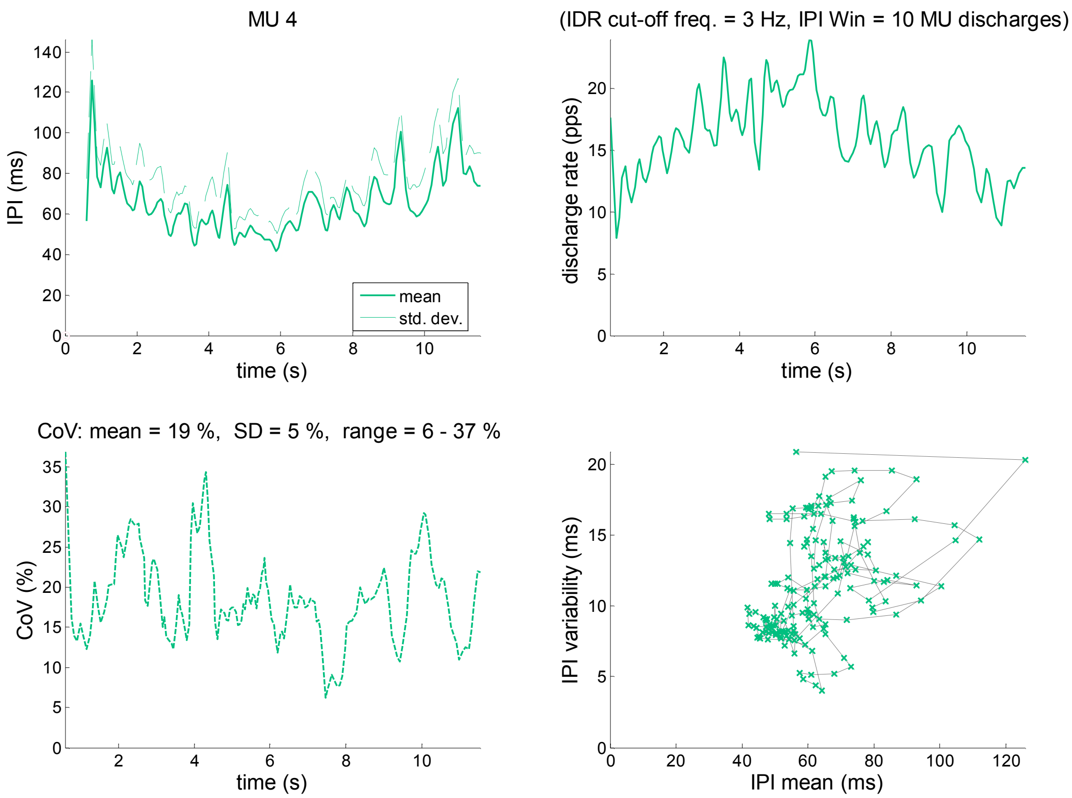

Figure 53: Plots of inter-pulse interval (IPI) of MU 4 (*upper left panel*), smoothed discharge rate (*upper right panel*), Coefficient of Variability (CoV) of IPI (*bottom left panel*) and IPI variability (CoV) versus IPI mean. Smoothed discharge rate is calculated by low-pass filtering the instantaneous discharge rate with cut-off frequency of 3 Hz. Standard deviation of IPI is calculated over 10 consecutive MU discharges.

4.7.4 Multichannel MUAP plots

Multichannel MUAP plots (so called MU fingerprints) can be plotted by clicking on a "`Plot MUAPs`" button (**Figure 54**, left panel). MU to be depicted is selected in a "`Selected MU`" pup-up menu (**Figure 54**, right panel). A matlab figure opens (**Figure 55**) with MUAP shapes as estimated by a spike triggered averaging of each acquired sEMG channel. Displayed MUAPs are spatially organized in rows and columns, reflecting the relative position of pick-up electrodes.

Figure 54: "`Plot MUAPs`" button (*left panel*) and "`Selected MU`" pup-up menu (*right panel*).

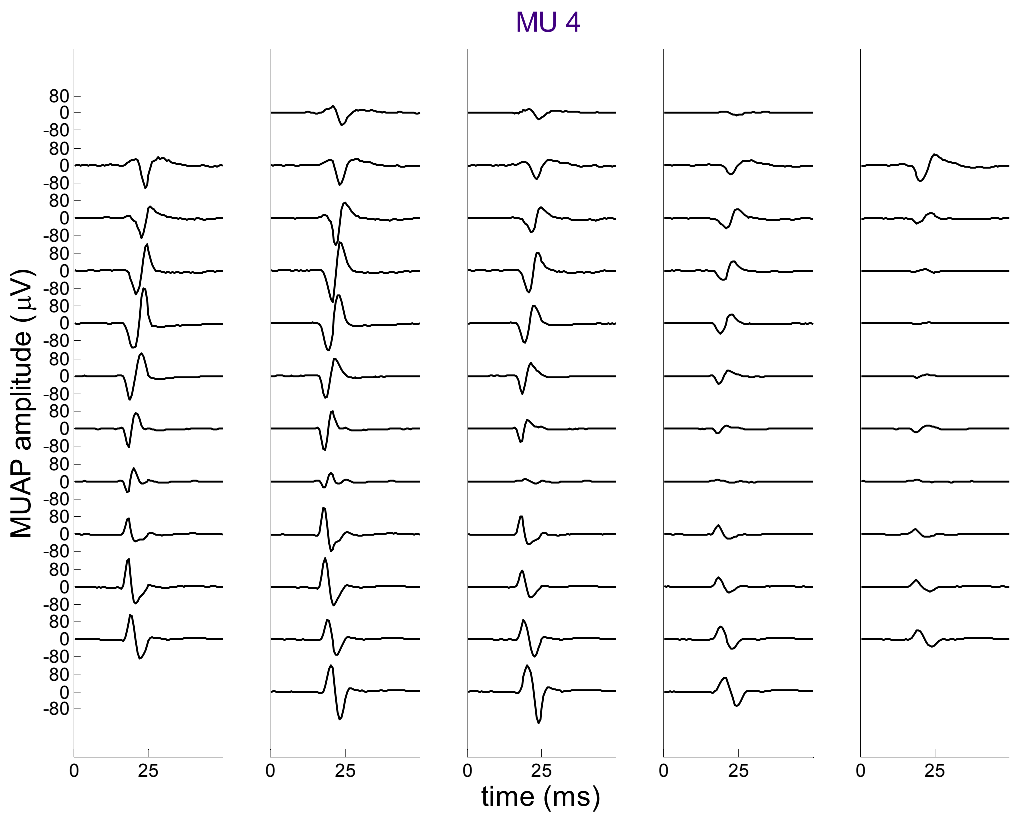

Figure 55: Multichannel MUAPs of MU 4 estimated by spike-triggered averaging of sEMG signals. sEMG signals were recorded with a grid of 61 electrodes arranged in 5 columns and 13 rows. The location of the innervation zone, tendon regions and propagation of motor unit action potentials are visible.

4.7.5 2D MUAP map animation

DEMUSEtool offers two animations of MUAP generation, propagation and attenuation process. The first one, so called 2D MUAP map, is a pseudocolor plot of the estimated MUAP amplitude in a given time instant. First, MUAP templates are estimated by a spike triggered averaging of the sEMG channels. sEMG channels are then spatially organized into a discrete 2D map, reflecting the relative position of pick-up electrodes. The amplitudes of MUAP templates at a given time instant specify the colour on this 2D map of channels. The missing intermediate points on the map are calculated by bilinear interpolation of the MUAP amplitudes in four adjacent sEMG channels. In the next animation frame, the animation time is advanced by one sample and the 2D MUAP map is recalculated (**Figure 57**).

Figure 56: "`Animate 2D MUAP map`" button (*left*), "`Selected MU`" pup-up menu and slider for selection of frame-rate (*right*)

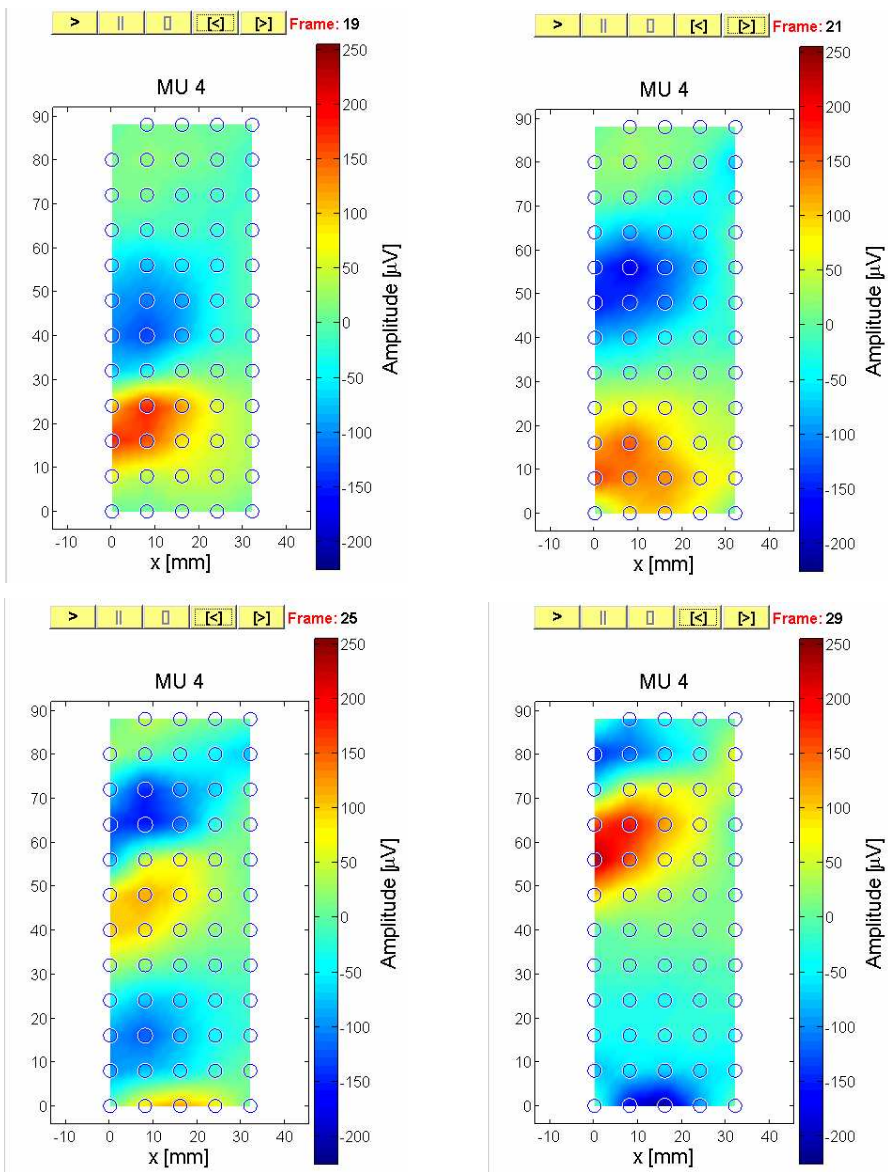

Figure 57: Pseudocolor animation of a MUAP generation, propagation and attenuation process. MUAP amplitudes on different sEMG channels (black & white circles) specify the colours of corresponding points on the 2D map.

Colours of intermediate map points are calculated by the bilinear interpolation of the MUAP amplitudes in four adjacent sEMG channels.

The MU and the animation frame rate are selected by the pup-up menu and slider shown in **Figure 56** (*right panel*). After clicking on the "`Animate 2D MUAP map`" button (**Figure 56**, *left panel*) animation window opens (**Figure 57**) and the animation starts automatically. The animation begins approx. 5 ms before the actual generation of the multi-channel MUAP and ends approx. 5 ms after the MUAP attenuation. During the animation, the propagation of MUAP along the muscle fibers can be observed (**Figure 57**).

Buttons on the top of the animation window (**Figure 57**) enable the following actions:

- ">" button: (re-)plays the animation;
- "||" button: pauses the animation;
- "[]" button: stops the animation;
- "[<]" button: animates the previous animation frame (i.e., step backward);
- "[<]" animates the next animation frame (i.e., step forward).

Current animation frame is displayed in the top right corner of the animation window (**Figure 57**).

Note: ! *Animation window cannot be closed while the animation is running. Stop the animation (by clicking on "||"or "[]"button) before you close the figure.*

4.7.6 3D MUAP map animation

The second animation, provided by the DEMUSEtool, includes a 3D plot of MUAP amplitude in time. By analogy with the animation of 2D MUAP map, MUAP templates are first estimated by a spike triggered averaging of sEMG channels. sEMG channels are then spatially organized into a discrete 2D map, reflecting the relative position of pick-up electrodes. The amplitudes of MUAP templates at a given time instant specify the height on this 2D map of channels. Missing intermediate points on the map are calculated by bilinear interpolation of MUAP amplitudes in four adjacent sEMG channels. In the next animation frame, the time is moved forward by one signal sample and the 3D map is recalculated.

To start the 3D animation, select the MU and the animation frame rate (**Figure 58** *right panel*). After clicking on the "`Animate 3D MUAP map`" button (**Figure 58**, *left panel*) the animation window opens (**Figure 59**) and the animation automatically starts. The animation begins approx. 5 ms before the actual generation of the MUAP and ends approx. 5 ms after the MUAP attenuation. During the animation, the propagation of MUAP along the muscle fibers can be observed (**Figure 59**).

Figure 58: "`Animate 3D MUAP map`" button (*left*), "`Selected MU`" pup-up menu and slider for selection of frame-rate (*right*)

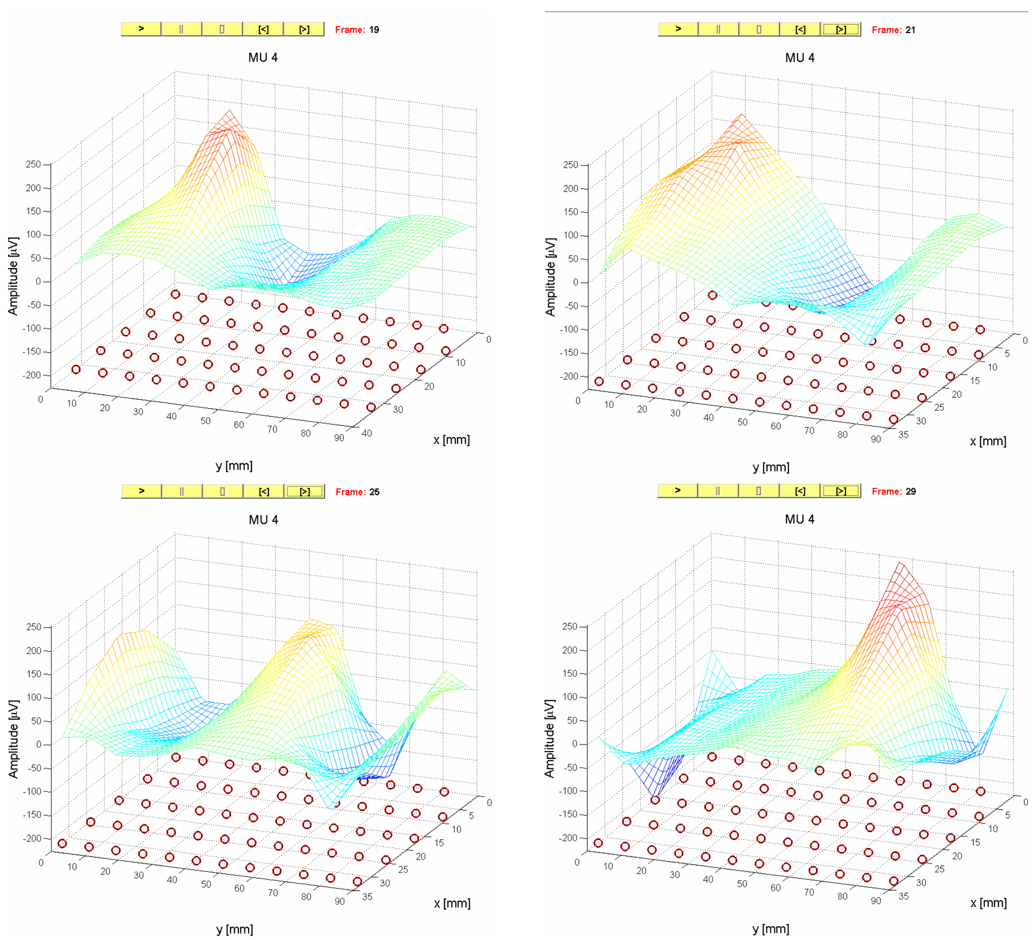

Figure 59: 3D animation of MUAP generation, propagation and attenuation. MUAP amplitudes on different sEMG channels (red circles) specify the height of corresponding points on a 2D map (heights of intermediate map points are calculated by the bilinear interpolation of the MUAP amplitudes in four adjacent sEMG channels).

Buttons on the top of the animation window (**Figure 59**) enable the following actions:

- “>” button: (re-)plays the animation;
- “||” button: pauses the animation;
- ”[]” button: stops the animation;
- “[<]” button: animates the previous animation frame (i.e., step backward);
- “[<]” animates the next animation frame (i.e., step forward).

The current animation frame is displayed in the top right corner of the animation window (**Figure 59**). The animation window is a regular matlab figure and can be freely manipulated by all available matlab graphical tools (e.g. zooming, coping, printing etc.).

Note: ***!*** *Animation window cannot be closed during the run of animation. Stop the animation (by clicking on “||”or ”[]”button) before you close the figure.*

During the 3D animation, the axes of the 3D plot can be freely rotated. To rotate a 3-D axes, click on the axes and drag the cursor in the direction you want to rotate. When you release the mouse button, DEMUSEtool redraws the axes in the new orientation (**Figure 60**).

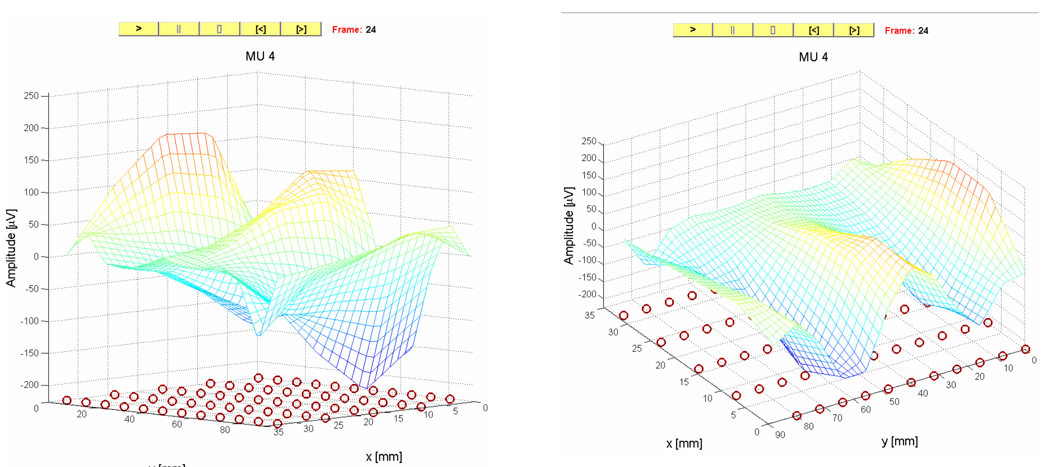

Figure 60: 3D rotation of the axes; the axes of the 3D plot can be rotated by dragging the cursor.

4.7.7 Plots of reconstructed MUAP trains

DEMUSEtool provides tool for plotting the sum of reconstructed MUAP trains superimposed to the original sEMG signals. This proves beneficial when evaluating the efficiency of the decomposition process. In surface EMG, there are many small and deep MUs which cannot be recognized. They contribute the background (physiological) noise. The second source of noise is so called instrumentation or thermal noise, which originates from the instrumentation's parasite capacities, line interference, etc. All together, these sources add to the measurement noise and affect the efficiency of the sEMG decomposition. By comparing the sum of the reconstructed MUAP trains to the original sEMG signal one can estimate the proportion of recognized sEMG components and, hence, speculate on the power (and even on the nature) of the measurement noise.

In the DEMUSEtool, the reconstructed MUAP trains are calculated as follow. Firstly, the MUAP shapes are estimated by spike triggered averaging of the acquired sEMG channel, using the identified MU discharge instants as triggers. The estimated MUAP shapes are then convolved with the identified MU discharge patterns and summed together. The sum of MUAP trains is subtracted from the original sEMG signals and the following signal-to-interference ratio (SIR) between the original sEMG signals and the residue after the subtraction is calculated:

$$SIR(i) = \left(1 - \frac{E\left[(x_i(n) - \sum_j z_{ij}(n))^2\right]}{E\left[x_i^2(n)\right]}\right) \cdot 100\%$$

where $x_i(n)$ denotes the *i*-th sEMG measurement, $z_{ij}(n)$ stands for the *j*-th MU's MUAP train reconstructed from the *i*-th sEMG measurement and E stands for sample mean. Finally, the range of SIRs of sEMG channels is displayed together with the reconstructed MUAP trains (**Figure 62**).

To display reconstructed MUAP trains, select the corresponding electrode row or electrode column (**Figure 61**, *left panel*) and click on "`Plot MUAP trains`" button (**Figure 61**, *right panel*). Matlab figure with selected sEMG channels and corresponding MUAP trains appears (**Figure 62**).

Note: ! *Due to the large number of acquired sEMG channels, only selected row/column of sEMG channels can be displayed in one figure. The number of figures, however, is not limited. You can display the MUAP trains on all the sEMG channels by consecutively selecting the different electrode columns, for example.*

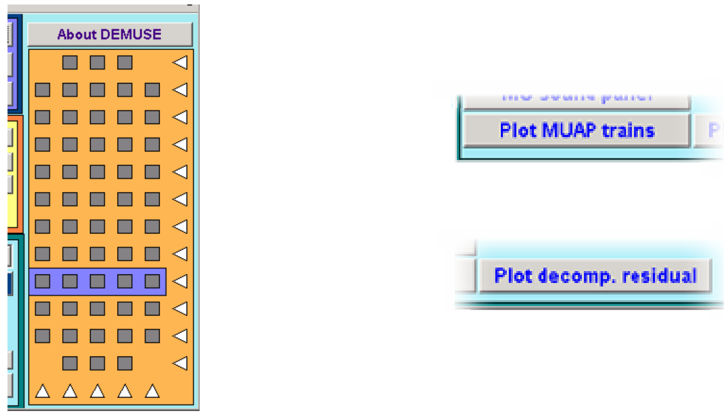

Figure 61: Channels selection frame (*left*), "`Plot MUAP trains`" button and "`Plot decomp. residual`" button (*right*).

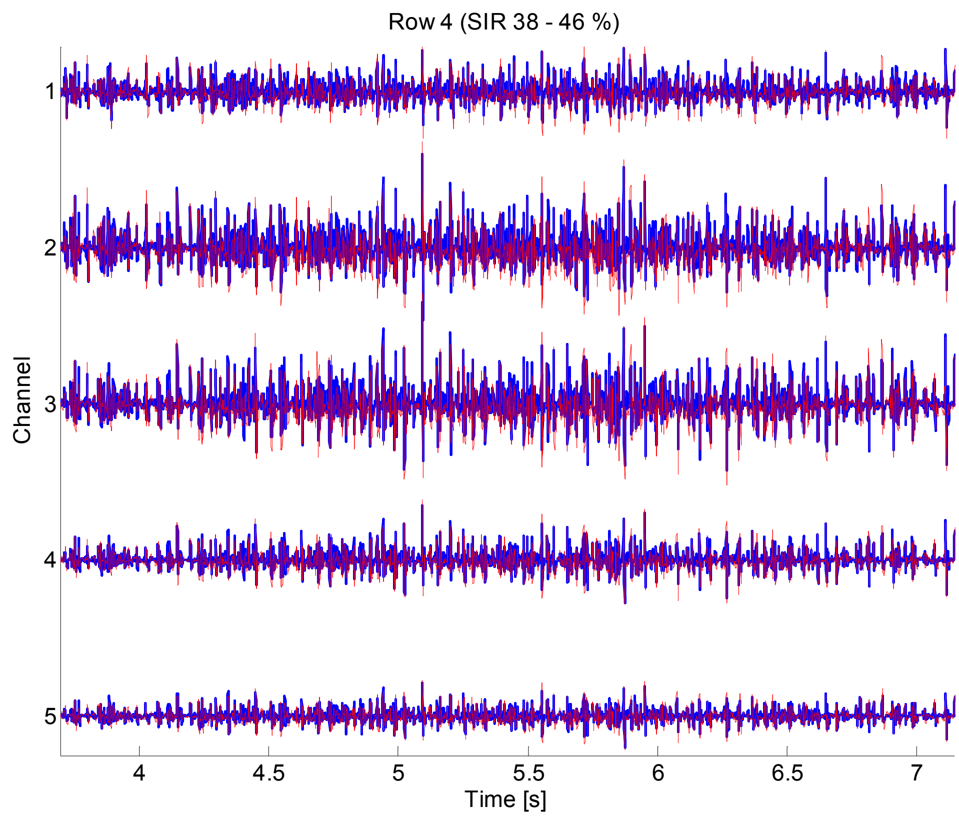

Figure 62: Matlab figure of selected sEMG channels (*blue lines*) and corresponding reconstructed MUAP trains (*red lines*). The range of SIRs of the depicted channels is displayed on the top of the figure.

Plots of reconstructed MUAP trains are displayed as matlab figures and can be freely manipulated by matlab figure editing tools (i.e., figure resizing, zooming, rotating, printing, etc.). Zoomed-in portion of **Figure 62** is depicted in **Figure 63**. The user is referred to matlab documentation for further details on the use of the matlab's graphic user interface.

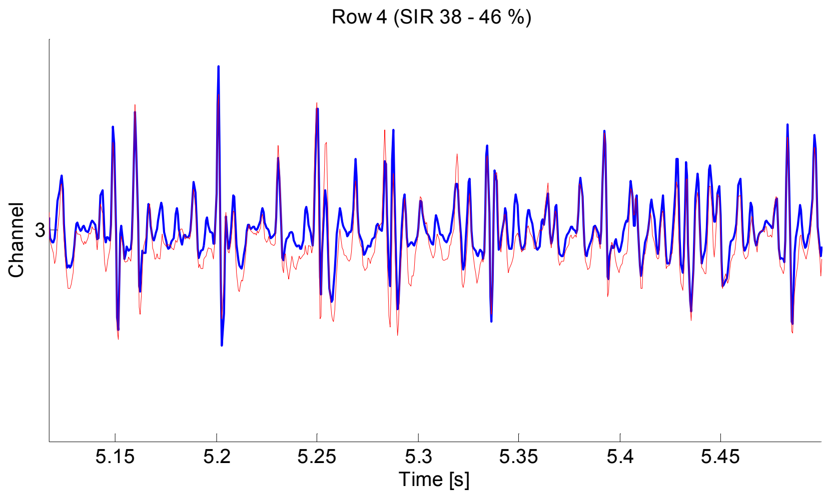

Figure 63: Matlab figure with selected sEMG channels and corresponding MUAP trains (short signal segment from the signal shown in **Figure 62**)

Click on "`Plot decomp. residual`" button (**Figure 61**, *left panel*) opens the Matlab figure with selected sEMG channels and corresponding residuals after subtraction of estimated MUAP trains (**Figure 64**). Range of SIRs of displayed sEMG channels is displayed at the top of the figure.

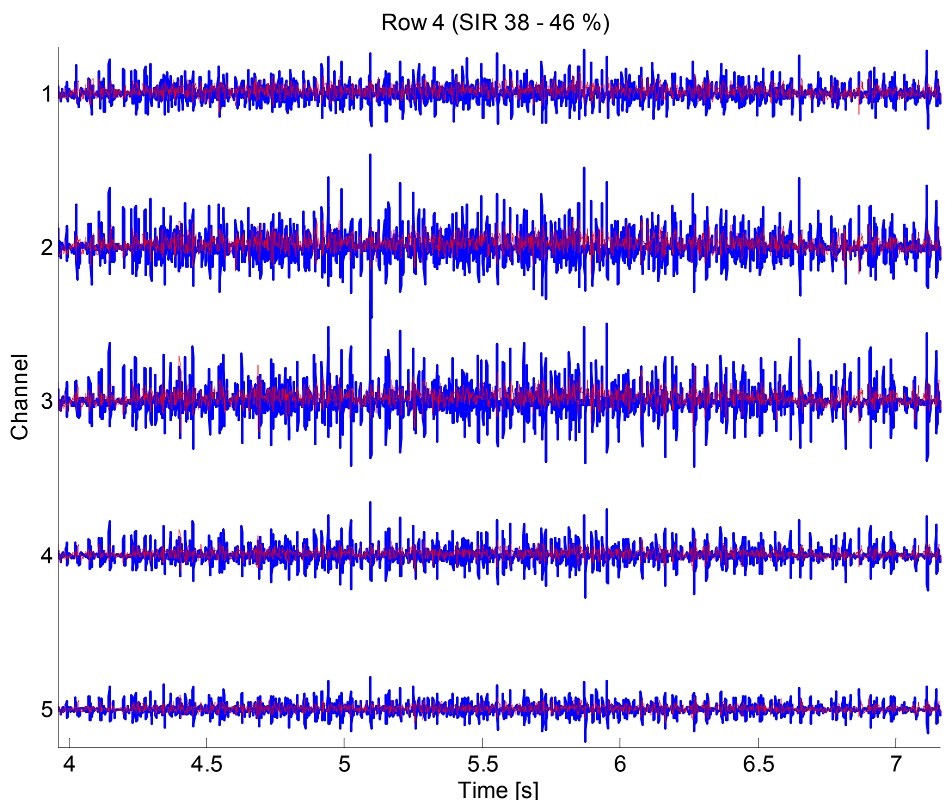

Figure 64: Matlab figure of selected sEMG channels (*blue lines*) and residual after subtraction of reconstructed MUAP trains (*red lines*). The range of SIRs of the depicted channels is displayed on the top of the figure.

4.7.8 MUAP statistics

DEMUSEtool integrates a simple tool for calculation of MUAP statistics of individual MU. Currently the following metrics are supported:

- MUAP peak-to-peak amplitude;
- MUAP energy;
- MUAP length;
- MU conduction velocity (CV);
- Correlation coefficient between the MUAP shapes used for calculation of MU conduction velocity;
- MU conduction velocity on a selected triplet of sEMG channels (triplet with highest correlation coefficient between the MUAP shapes is automatically selected for this metrics).

All these metrics are calculated over double-differential spatial derivations of a raw sEMG signals. MUAP peak-to-peak amplitude is defined as a distance between the highest positive peak and the lowest negative peak of MUAP, MUAP energy is calculated as a Root Mean Square of a MUAP, whereas MUAP length is defined as a maximal time distance between the points in which rectified MUAP surpasses the 5 % of its maximal amplitude. Conduction velocity is estimated with the multi-channel algorithm described by Farina et al. [8] from triplets of double differential derivations. Triplet with highest correlation coefficient between the MUAP shapes (calculated pair-wise) is automatically selected for calculation of MU conduction velocity on a selected triplet of sEMG channels.

MUAP statistics tool (**Figure 66**) opens after clicking on "`calc MUAP stat`" button (**Figure 65**). Upper panel displays MUAP shapes as estimated by a spike triggered averaging of each of double-differential derivatives. Displayed MUAPs are spatially organized in rows and columns, reflecting the relative position of pick-up electrodes.

Lower panel displays the MUAP statistics (**Figure 66**). Left-bottom window displays statistics for individual MU averaged over all surface channels. Right-bottom window reports statistics averaged over all the MUs. Mean value, standard deviation, and minimum and maximum values are reported for each of aforementioned metrics. Triplet of sEMG channels, selected for calculation of "`MU conduction velocity on selected channels`" is denoted by red triangles. User can move among different MUs by clicking on '<<' and '>>' buttons.

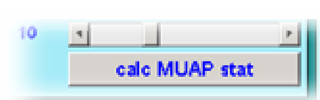

Figure 65: "`calc MUAP stat`" button.

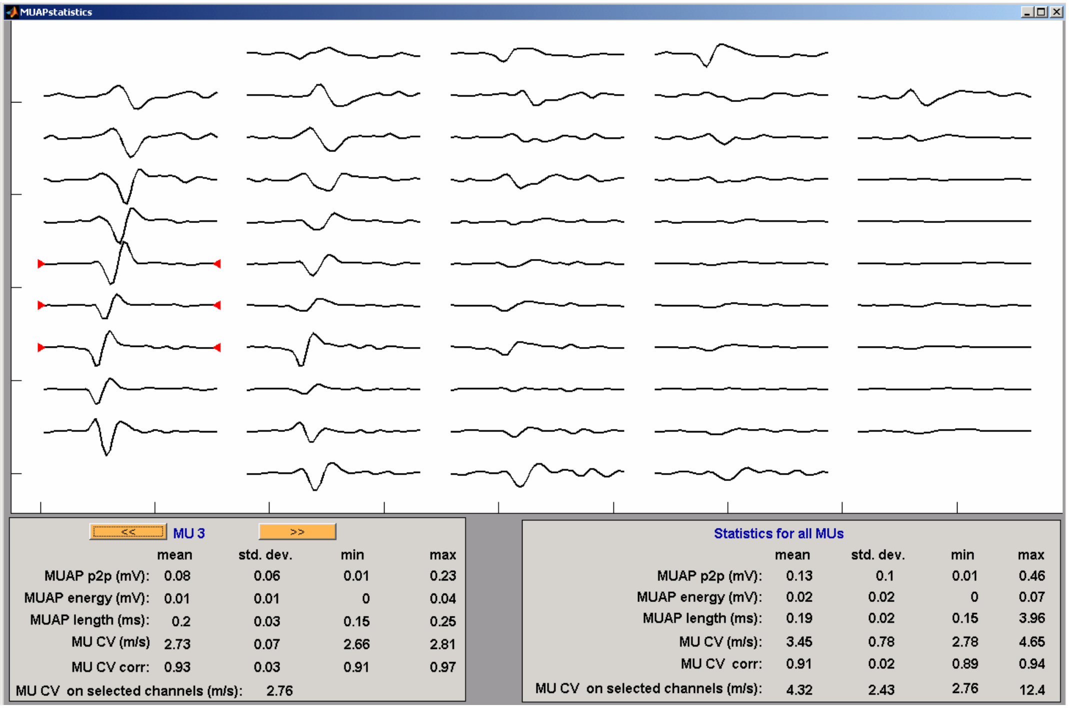

Figure 66: "`MUAP statistics`" window with upper panel displaying multichannel MUAP of selected MU (as spatial double-differential derivatives of sEMG channels). Displayed MUAPs are spatially organized in rows and columns, reflecting the relative position of pick-up electrodes. Lower panel displays the MUAP peak-to-peak amplitude (MUAP p2p), MUAP energy, MUAP length, MU conduction velocity (MU CV), correlation coefficient between the MUAP shapes used for calculation of MU conduction velocity (MU CV corr) and MU conduction velocity on a selected triplet of sEMG channels (triplet denoted by red triangles in the upper panel). Left-bottom window displays statistics for an individual MU averaged over all surface channels. Right-bottom window reports statistics averaged over all the MUs. Mean value, standard deviation, minimum and maximum are reported for each of aforementioned metrics. User can move among different MUs by clicking on '<<' and '>>' buttons.

4.8 Saving and reloading of the decomposition results

Decomposition results can be saved by clicking on the “`Save results`” button (**Figure 67**). The results are automatically saved into the directory containing the currently loaded SIG file. The following file naming convention is used:

`NameOfTheSIGfile_offsetN_runsM.mat`

the name of the currently loaded SIG file	Initial signal offset	Number of decomposition iterations

where `NameOfTheSIGfile` stands for the name of the currently loaded SIG file, `N` is the initial signal offset (in seconds) and `M` is the number of decomposition runs (see Section 4.4 for details). For example, the decomposition results of a SIG file `Subject1.SIG` with initial signal offset set equal to 0 and number of decomposition iterations set equal to 30 is saved in the following matlab file:

`Subject1_offset0_runs30.mat`

Saved results can be reloaded by clicking on the “`Load results`” button (**Figure 67**). “Load results” dialog window opens (**Figure 68**). Choose the `*.mat` file and click on “`Open`” button. Once reloaded into the DEMUSEtool, results can be freely edited and displayed (graphical representations and animations of the reloaded results are fully supported). “`Save results`” button saves all the decomposition results, including the original sEMG singals.

Figure 67: “`Save results`” button (*left*) and “`Load results`” button (*right*).

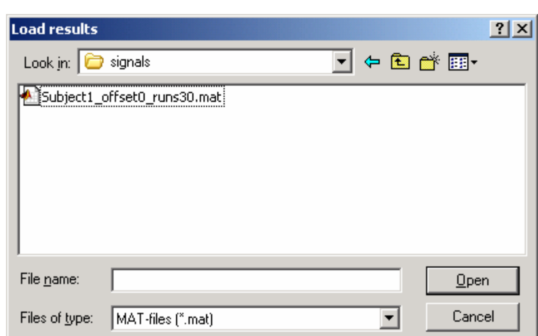

Figure 68: “`Load results`” dialog window.

To save just the discharge patterns of MUs, click on a "`Save MU discharges`" button (**Figure 69**). MU discharges are automatically saved into the directory containing the currently loaded SIG file. The following file naming convention is used:

`NameOfTheSIGfile_MUPulses_offsetN_runsM.mat`

the name of the currently loaded SIG file

Initial signal offset

Number of decomposition iterations

where `NameOfTheSIGfile` stands for the name of the currently loaded SIG file, `N` is the initial signal offset (in seconds) and `M` is the number of decomposition runs (see Section 4.4 for details). MU discharges are saved into a Matlab cell structure `MUPulses`, with discharge times (in samples) of single MU in each cell. For example, discharge times of MU 1 are stored in cell `MUPulses{1}`, discharge times of MU 2 in cell `MUPulses{2}` etc.

Figure 69: "`Save MU discharges`" button (*left*) and "`Load MU discharges`" button (*right*).

4.9 DEMUSEtool acknowledgements

Information about the DEMUSEtool version, copyrights and author's acknowledgement are displayed by clicking on the "`About DEMUSE`" button (**Figure 70**). "`About DEMUSE`" dialog window opens (**Figure 71**).

Figure 70: "`About DEMUSE`" button.

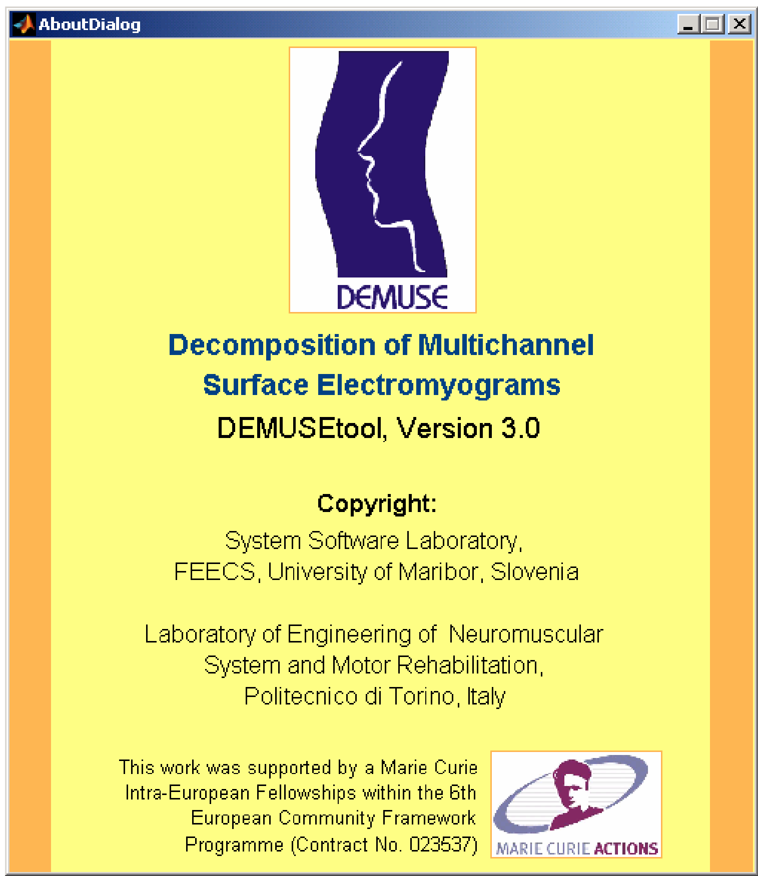

Figure 71: "`About DEMUSE`" dialog window.

5 Appendix I: definition of DEMUSEtool reader

When loading the sEMG files, DEMUSEtool prompts for selection of a proper reader (**Figure 6**). There are several readers already implemented in DEMUSEtool, supporting all main acquisition systems currently available in LISiN lab. For more complex acquisition configurations or other acquisition systems, specialized reader of the sEMG files can be implemented and added to the `DEMUSE_readers` directory (located in the main directory of the DEMUSEtool). The name of the reader must start with the string "`DEMUSEtool_reader`", but can continue with arbitrary name. A good practice is to specify the main parameters of the reader in its name, so the user can easily identify it. For example, reader

```
DEMUSEtool_reader_5x13_IED5mm_telescopic_pins_SD.m
```

denotes the reader which reads surface EMG acquired by a matrix of 5x13 electrodes with inter-electrode distance of 5 mm and electrodes on telescopic pins. EMG signals were acquired in single-differential (SD) mode.

DEMUSEtool automatically loads all the files whose name starts with "`DEMUSEtool_reader`" into the list of available readers and display their descriptions in the `Reader Dialog Window`. Reader for specific files can also be specified in a text file. Simply write the name of corresponding Matlab routine in the text file called `DEMUSE_reader_Tag.txt`, e.g.:

```
DEMUSEtool_reader_5x13_IED5mm_telescopic_pins_SD.m
```

and save the file into the directory with corresponding sEMG files. DEMUSEtool will automatically check the directory with sEMG files for `DEMUSE_reader_Tag.txt` file and, if found, use the reader specified therein for all the sEMG files in the corresponding directory.

Structure of reader is exemplified in **Figure 72**. All the text before the reserved keyword '`INPUTS`' is considered as a description and displayed by DEMUSEtool in the `Reader Dialog Window` (**Figure 6**).

Inputs to the reader are limited to the path and name of the sEMG file and (optionally) the length of the signal to be loaded (in seconds). If no signal length is specified, the reader should return the entire signal in the file.

```
function data = DEMUSEtool_reader_5x13_5mm_telescopic_pins_SD_4amplifiers
          (filepath,filename,epoch_length)
% function data = DEMUSEtool_reader_5x13_5mm_telescopic_pins_SD_4amplifiers
%                      (filepath,filename,epoch_length)
%
% DEMUSEtool reader: reads surface EMG acquired by a matrix of 5x13
% electrodes with inter-electrode distance of 5 mm and telescopic pins.
% Acquisition board for 64 channels is made by synchronization of 4
% 16-channel amplifiers.
%
% INPUTS:
%   - filepath: directory with the SIG file to be loaded
%   - finename: SIG file to be loaded
%   - epoch_length: (optional) length of the epoch of signal to be loaded (in s)
%
% OUTPUT:
%   data: structure with the following fields;
%      SIG - two dimensional cell array with surface EMG channel in each
%           cell - SIG{r,c} is the channel in row r and column c. Missing
%           electrodes are denoted by empty arrays, e.g. SIG{1,1} = [];
%      fsamp - sampling frequency of sEMG
%      signal_length - length of a surface EMG signals (in samples)
%      montage - montage of electrodes - 'MONO' for monopolar, 'SD' for
%              single differential
%      IED - inter-electrode distance (in mm)
%      force - measured force signal if avalable, empty array otherwise
%      AUXchannels - auxilary channels (currently not used by DEMUSEtool
%      AUXchannels_description - cell array of texts describing the data
%               in AUXchannels (one cell per channel)
%
% ----------------------------------------------------------------------
% Copyright: LISiN, Politecnico di Torino, Italy
%          SSL, FEECS, University of Maribor, Slovenia
% Author: Ales Holobar (ales.holobar@uni-mb.si)
% Last modified: 14. 10. 2008
```

description of a reader (anything before the keyword INPUT will appear in the reader description window)

Inputs to the reader are standardized.

Outputs are always in the form of Matlab structure.

Figure 72: Reserved head of a sEMG reader and its standardized interface. All the DEMUSEtool readers must specify this initial structure and must use specified inputs and outputs. Actual implementation of the loading method is left to the user.

Outputs of the reader are given in the form of Matlab structure with the following field requested:

- **SIG**: two dimensional cell array with surface EMG channel in each cell - `SIG{r,c}` is the channel in row r and column c. Missing electrodes are denoted by empty arrays, e.g. `SIG{1,1} = []`.
- **fsamp**: sampling frequency of sEMG.
- **signal_length**: length of a surface EMG signals (in samples).
- **montage**: montage of electrodes - 'MONO' for monopolar, 'SD' for single differential configuration.
- **IED**: inter-electrode distance (in mm).
- **force**: measured force signal if available, empty array otherwise.
- **AUXchannels**: auxilary channels (currently not used by DEMUSEtool.
- **AUXchannels_description**: cell array of texts describing the data in AUXchannels (one cell per channel).

6 Appendix II: gradient Convolution Kernel Compensation

Gradient Convolution Kernel Compensation (gCKC) technique [3] is fully automatic, resolves MUAP superimpositions, and relies minimally on anatomic properties of the investigated muscle. Moreover, it implicitly combines all the available information provided by all the sEMG measurements. By compensating for the shapes of the detected MUAPs, it directly estimates MU discharge patterns without reconstructing the detected MUAP shapes. This significantly decreases the number of unknowns to be estimated and reduces computational time. MUAP shapes can then be estimated by spike triggered averaging of sEMG measurements.

When compared to other currently available surface EMG decomposition techniques, gCKC exhibits high accuracy, efficiency and robustness in identification of MU discharge patterns and is specifically tailored to low-quality noisy signals. This extension is of paramount importance for clinical practice, where recoding environment cannot be strictly controlled. The accuracy obtained with the gCKC decomposition technique is comparable to that obtained by decomposition of intramuscular recordings [9].

In the DEMUSEtool, reconstructed MU discharge patterns are automatically tested against the predefined ranges of physiological variables (i.e., discharge rate, variability of inter-pulse interval, etc.) and sorted with respect to the estimated degree of decomposition reliability, with MU 1 having the highest decomposition reliability.

In the sequel, technical details of gradient CKC method are briefly summarized. Text provided describes the main decomposition concepts only, whereas theoretical derivations of mathematical formulas are skipped. Interested reader is referred to manuscripts [3] and [4] for all the details.

6.1 Data model

Suppose the sEMG signals are observed in *M* detection points and denote their sampled vector as $\mathbf{x}(n) = [x_1(n),....,x_M(n)]^T$, where $x_i(n)$ stands for the *n*-th sample of the *i*-th measurement. In the case of isometric muscle contractions, the measurements $\mathbf{x}(n)$ can be modelled as outputs of a linear time-invariant (LTI) multiple-input multiple-output (MIMO) system [4]:

$$x_i(n) = \sum_{j=1}^{N} \sum_{l=0}^{L-1} h_{ij}(l) t_j(n-l) + \omega_i(n), \quad i=1,...,M. \tag{1}$$

where $\omega_i(n)$ stands for zero-mean additive noise samples. Each model input $t_j(n)$ is considered a sample of a motor unit innervation pulse train (IPT), while causal and finite channel response $\mathbf{h}_{ij}=\{h_{ij}(l);\ l=0,1,\ldots,L\text{-}1\}$, corresponds to the j-th, L samples long motor unit action potential, as detected by the i-th measurement. No a priori limitations are posed on the channel responses $\mathbf{h}_{ij}$. Thus, properties of the detected MU (e.g., its depth in the muscle tissue, action potential propagation velocity), as well as the properties of the detection system are described by $\mathbf{h}_{ij}$. The innervation pulse trains $\mathbf{t}(n)=[t_1(n),....,t_N(n)]^T$ represent MU discharge times only, and are modelled as sequences of Dirac impulses:

$$t_j(n)=\sum_{k=-\infty}^{\infty}\delta\left[n-T_j(k)\right],\ j=1,...,N \tag{2}$$

where the k-th MUAP of the j-th MU appears at time $T_j(k)$.

Eq. (1) can be rewritten in a matrix form:

$$\mathbf{x}(n)=\mathbf{H}\overline{\mathbf{t}}(n)+\boldsymbol{\omega}(n) \tag{3}$$

where $\boldsymbol{\omega}(n)=[\omega_1(n),....,\omega_M(n)]^T$ is a noise vector and the mixing matrix $\mathbf{H}$ comprises all the MUAPs as detected by the surface electrodes,

$$\mathbf{H}=\begin{array}{l}\begin{bmatrix} h_{11}(0) & \cdots & h_{11}(L-1) & h_{12}(0) & \cdots & h_{12}(L-1) & \cdots \\ h_{21}(0) & \cdots & h_{21}(L-1) & h_{22}(0) & \cdots & h_{22}(L-1) & \cdots \\ \vdots & & \vdots & \vdots & & \vdots & \cdots \\ h_{M1}(0) & \cdots & h_{M1}(L-1) & h_{M2}(0) & \cdots & h_{M2}(L-1) & \cdots \end{bmatrix} \begin{array}{l}\}\textbf{MUAPs in measurement 1}\\ \}\textbf{MUAPs in measurement 2}\\ \vdots \\ \}\textbf{MUAPs in measurement M}\end{array}\\ \qquad\underbrace{\qquad\qquad\qquad\qquad}_{\textbf{MUAPs of MU1}}\ \underbrace{\qquad\qquad\qquad\qquad}_{\textbf{MUAPs of MU2}}\end{array}$$

while the vector

$$\overline{\mathbf{t}}(n)=\left[t_1(n),t_1(n-1),...,t_1(n-L+1),...,t_N(n),...,t_N(n-L+1)\right]^T \tag{4}$$

stands for an extended form of $\mathbf{t}(n)=\left[t_1(n),...,t_N(n)\right]^T$. For decomposition, it is beneficial to further extend the vector $\mathbf{x}(n)$ by *K-1* delayed repetitions of each measurement:

$$\overline{\mathbf{x}}(n)=[x_1(n),x_1(n-1),....,x_1(n-K+1),....,x_M(n),....,x_M(n-K+1)]^T. \tag{5}$$

This increases the number of observations with respect to the unknowns in *Eq.* (3). The optimal value of the extension factor K is application dependent and is typically between 5 and 15. This factor is automatically selected by the DEMUSEtool.

6.2 Decomposition method

The gCKC method fully automates the identification of MU discharge sequences in the *Eq.* (3). In the first step, the method blindly estimates the cross-correlation vector $\mathbf{c}_{t_j\mathbf{x}} = E\left(t_j(n)\overline{\mathbf{x}}^T(n)\right)$ between the *j*-th pulse train and the extended measurements, where *E*(.) stands for mathematical expectation. In the second step, the unknown mixing matrix $\mathbf{H}$ is compensated by calculating the linear minimum mean square error (LMMSE) estimator of the *j*-th pulse train t_j :

$$\hat{t}_j(n) = \mathbf{c}_{\mathbf{t}_j\overline{\mathbf{x}}}\mathbf{C}_{\overline{\mathbf{x}}\overline{\mathbf{x}}}^{-1}\overline{\mathbf{x}}(n) = \mathbf{c}_{\mathbf{t}_j\overline{\mathbf{t}}}\mathbf{H}^T\left(\mathbf{H}\mathbf{C}_{\overline{\mathbf{t}}\overline{\mathbf{t}}}\mathbf{H}^T + \mathbf{C}_{\boldsymbol{\omega}\boldsymbol{\omega}}\right)^{-1}\mathbf{H}\overline{\mathbf{t}}(n)\,. \tag{6}$$

where $\mathbf{C}_{\overline{\mathbf{x}}\overline{\mathbf{x}}} = E\left(\overline{\mathbf{x}}(n)\overline{\mathbf{x}}^T(n)\right)$ is the correlation matrix of extended measurements $\overline{\mathbf{x}}(n)$, and $\mathbf{c}_{t_j\overline{\mathbf{t}}} = E\left(t_j(n)\overline{\mathbf{t}}^T(n)\right)$ is the vector of cross-correlation coefficients between the IPT of the *j*-th MU and the IPTs of all the MUs active in the detection volume.

Estimator (6) requires the cross-correlation vector $\mathbf{c}_{\mathbf{t}_j\overline{\mathbf{x}}}$ to be known in advance. This is never the case and Holobar and Zazula proposed gradient-based procedure for its blind estimation [4].

Let $\overline{F}(t_j) = \sum_m F(t_j(m))$ denote a cost function in the space of MU discharge sequences, with arbitrary differentiable scalar function $F(t)$ applied to each sample of the estimated pulse sequence $t_j(n)$. General iteration step for gradient optimization of $\mathbf{c}_{\mathbf{t}_j\overline{\mathbf{x}}}$ is then defined as:

$$\mathbf{c}_{t_j\mathbf{y}}^{\{k+1\}} = \mathbf{c}_{t_j\mathbf{y}}^{\{k\}} + \eta(k)\sum_m \frac{\partial F(t_j(m))}{\partial t_j(m)}\mathbf{y}(m) \tag{7}$$

where $\eta(k)$ is the learning rate in the *k*-th iteration step.

Denote $f(t) = \partial F(t)/\partial t$. Then the second factor in the update rule (7) simplifies to

$$\sum_m f(\hat{t}_j(m))\overline{\mathbf{x}}(m) = \mathbf{H}\mathbf{C}_{\overline{\mathbf{t}}\overline{\mathbf{t}}}\sum_{k_2=0}^{\infty}\frac{\prod_{p=1}^{k_2}(2p-1)}{(2k_2)!}\sigma_b^{2k_2}\boldsymbol{f}^{(2k_2)}\left(\mathbf{d}^{\{k\}}\right) \tag{8}$$

where $\mathbf{d}^{\{k\}} = \mathbf{H}^T\mathbf{C}_{\overline{\mathbf{y}}\overline{\mathbf{y}}}^{-1}\hat{\mathbf{c}}_{t_j\mathbf{y}}^{\{k\}}$ is a match filter of the j-th MU in the space of MU discharge patterns, σ_b is standard deviation of $b(m) = \mathbf{d}_0^T\mathbf{H}^{-1}\overline{\boldsymbol{\omega}}(m)$ and $\boldsymbol{f}^{(k)}$ stands for a vector function with the *r*-th element defined as $\boldsymbol{f}^{(n)}(\mathbf{d})\big|_r = f^{(n)}(d_r)$, where $f^{(n)}(t)$ is the *n*-th derivative of $f(t)$.

When $\sigma_b^2 < 1$ and $f^{(2k_2)}(t)$ are limited on the interval [-1,+1], higher order terms of expansion in (8) decay rapidly and the first approximation of the update rule (7) yields

$$\hat{\mathbf{c}}_{t_j\mathbf{y}}^{\{k+1\}} \approx \hat{\mathbf{c}}_{t_j\mathbf{y}}^{\{k\}} + \eta(k)\mathbf{H}\mathbf{C}_{\mathbf{tt}}\boldsymbol{f}\left(\mathbf{H}^T\mathbf{C}_{\overline{\mathbf{yy}}}^{-1}\hat{\mathbf{c}}_{t_j\mathbf{y}}^{\{k\}}\right) \tag{9}$$

which, after using $\mathbf{d}_0^{\{k+1\}} = \mathbf{H}^T\mathbf{C}_{\overline{\mathbf{yy}}}^{-1}\hat{\mathbf{c}}_{t_j\bar{\mathbf{y}}}^{\{k+1\}}$, finally yields

$$\mathbf{d}_0^{\{k+1\}} = \mathbf{d}_0^{\{k\}} + \eta(k)\mathbf{H}^T\mathbf{C}_{\overline{\mathbf{yy}}}^{-1}\mathbf{H}\mathbf{C}_{\overline{\mathbf{tt}}}\mathbf{f}\left(\mathbf{d}_0^{\{k\}}\right) \tag{10}$$

Now, assume match filter $\mathbf{d}^{\{k\}}$ has converged to $\mathbf{d}^{\{k\}} = \mathbf{e}_j + \boldsymbol{\varepsilon}$, where $\mathbf{e}_j=[0,\ldots 0,1,0,\ldots 0]$ is unity vector with element at the *j*-th position equal to 1 and $\boldsymbol{\varepsilon}$ is vector of errors, $\|\boldsymbol{\varepsilon}\| \ll 1$. Using the Taylor expansion of $f(t)$, the updates of the *j*-th and the *i*-th rows (*i≠j*) in (9) yield:

$$\begin{aligned}\mathbf{f}\left(\mathbf{e}_j + \boldsymbol{\varepsilon}\right)\Big|_i &= f(0+\varepsilon_i) = f(0) + \textstyle\sum_{k=1}^{\infty}\frac{1}{k!}f^{(k)}(0)\varepsilon_i^k \\ \mathbf{f}\left(\mathbf{e}_j + \boldsymbol{\varepsilon}\right)\Big|_j &= f(1+\varepsilon_j) = f(1) + \textstyle\sum_{k=1}^{\infty}\frac{1}{k!}f^{(k)}(1)\varepsilon_j^k\end{aligned} \tag{11}$$

According to (11), in the vicinity of local optimum, convergence and stability of (10) depend on the values of the first few derivatives of $f(t)$ at points $t=0$ and $t=1$. The values of the first few derivatives of cost functions $f_1(t)=tanh(t)$, $f_2(t)=t\exp(-t2/2)$ and $f_3(t)=\log(1+t^2)$ are depicted in **Figure 73.** Functions $f_1(t)$, $f_2(t)$ were proposed by Hyvärinen [5] and are implemented in the popular fastICA algorithm. Function $f_3(t)$ was selected empirically and exhibits high robustness to outliers. Among these three criteria functions, $f_3(t)=\log(1+t^2)$ is least sensitive to errors in vicinity of 0 while its performance in vicinity of 1 is comparable to other two functions. Ideally, $\mathbf{d}^{\{k\}}$ has a large number of zeros (of the order of several hundreds) and a single value equal to 1. This makes low sensitivity to errors in vicinity of zeros a crucial advantage of $f_3(t)=\log(1+t^2)$ function. This is further demonstrated in **Figure 74**, where the value of vector $\mathbf{d}^{\{k\}}$ after the fifth iteration as calculated from theoretical approximation (9) (red line) and by direct numerical calculation (8) (blue line) is depicted. Compare the vector values close to zero among different functions $f_1(t)$, $f_2(t)$ and $f_3(t)$.

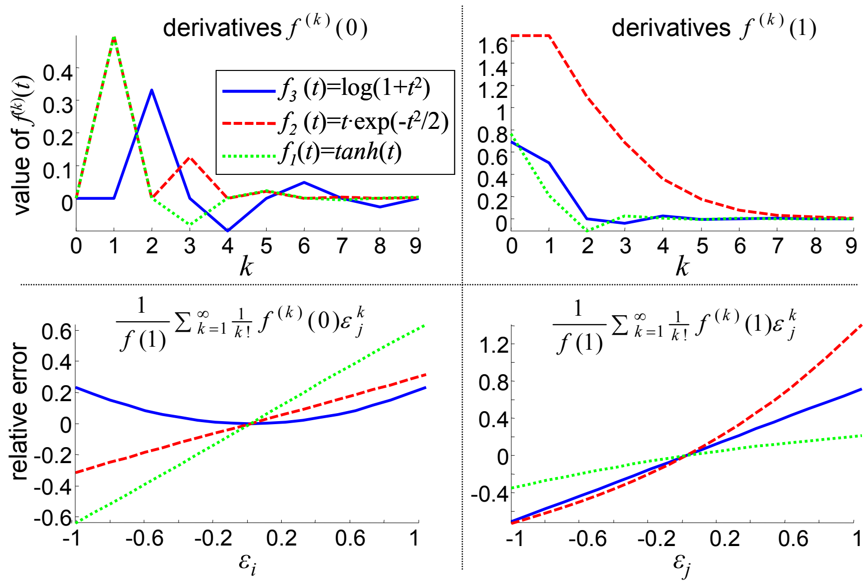

Figure 73: The first few derivatives of the cost functions $f_1(t)$, $f_2(t)$ and $f_3(t)$ at points $t=0$ (top left panel) and $t=1$ (top right panel), and the relative error contribution, as defined by (11), to $f(0)$ (bottom left panel) and to $f(1)$ (bottom right panel) for different values ε_j.

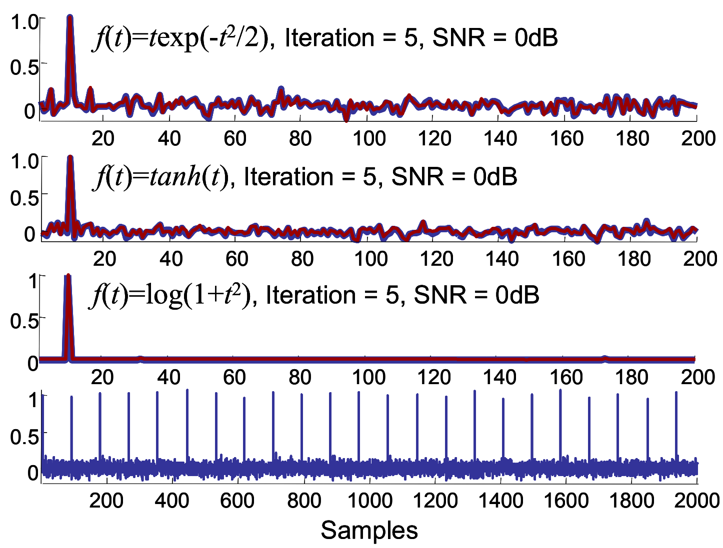

Figure 74: *Top three panels:* vector $\mathbf{d}^{\{k\}}$ after the fifth iteration as calculated from theoretical approximation (9) (red line) and by direct numerical calculation (8) (blue line); *Bottom panel*: estimated PS t after the fifth iteration of $f_3(t)=\log(1+t^2)$.

7 Technical support

DEMUSEtool is copyrighted by the Laboratory of Engineering of Neuromuscular System and Motor Rehabilitation (LISiN) from Politecnico di Torino, Italy, and System Software Laboratory (SSL) from University of Maribor, Slovenia. Its development was supported by a Marie Curie Intra-European Fellowship within the 6th European Community Framework Programme (DE MUSE, Contract No. 023537).

For further technical assistance and support, please contact

ales.holobar@delen.polito.it

or

ales.holobar@uni-mb.si

This work was supported by a Marie Curie Intra-European Fellowship within the 6th European Community Framework Programme (DE MUSE, Contract No. 023537).

References

1. Matlab, the language of technical computing, MathWorks Inc., web address: http://matworks.com

2. Acquisition Software, User Manual v1.62, OT Bioelettronica, SIRIO Automazione srl and LISiN – Bioengineering Center Politecnico di Torino, February 2007.

3. A. Holobar, D. Zazula: Gradient Convolution Kernel Compensation Applied to Surface Electromyograms, ICA 2007, LNCS 4666, pp. 617–624, 2007.

4. Holobar A and Zazula D: Multichannel Blind Source Separation Using Convolution Kernel Compensation, IEEE Trans. Sig. Process. 55 (9), 4487-4496, 2007.

5. Hyvärinen A: Fast and Robust Fixed-Point Algorithms for Independent Component Analysis, IEEE Trans. on Neural Networks, vol. 10, 1999, pp. 626-634.

6. EMG-USB electromyographic signal amplifier, User Manual v.1.32, OT Bioelettronica, SIRIO Automazione srl and LISiN – Bioengineering Center Politecnico di Torino, September 2006.

7. McGill KC, Lateva ZC, Marateb HR. EMGLAB: an interactive EMG decomposition program. J Neurosci Methods 2005; 149:121-133.

8. Farina D, Muhammad W, Fortunato E, Meste O, Merletti R, Rix H. Estimation of single motor unit conduction velocity from the surface EMG signal detected with linear electrode arrays. Med Biol Eng Comput. 2001; 39: 225-236.

9. A. Holobar, D. Farina, M. Gazzoni, R. Merletti, D. Zazula: Estimating Motor Unit Discharge Patterns from High-Density Surface Electromyogram, Clinical Neurophysiology, in press.