EMG Driven muscle Force Estimator (EMG-FE)

Version 1.0r7 distribution 1

User Guide

Luciano Luporini Menegaldo

Liliam Fernandes de Oliveira

Biomedical Engineering Department - COPPE
Federal University of Rio de Janeiro

e-mail: lmeneg@ufrj.br
www.peb.ufrj.br

Rio de Janeiro, Brazil

February 11, 2014
About

This software (or MATLAB toolbox) has been designed to facilitate the estimation of muscle forces from EMG signals. It guides the user, step by step, towards muscle force estimation beginning from raw EMG signals.

The Graphical User Interface (GUI) of this software has been implemented by the IME-Junior - Junior Enterprise of the Military Institute of Engineering, Rio de Janeiro, Brazil. www.imejunior.com.br, e-mail: contato@imejunior.com.br.

The authors are gratefully acknowledged to CAPES (Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior), to FAPERJ (Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), for financial support.

If you have any comment, bug report, suggestion, doubts etc. please send an e-mail to the authors.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses.
Contents

1 Introduction 4
   1.1 Signal collecting recommendations ................................. 5

2 Usage of the program 5
   2.1 Running the program .................................................. 5
   2.2 Choosing a display language ......................................... 6
   2.3 Creating a new test subject ........................................... 6
   2.4 Adding an MVC test .................................................... 7
      2.4.1 Filtering test data ............................................... 8
      2.4.2 MVC analysis .................................................... 9
   2.5 Adding the main 'Submaximal’ test ................................. 9
   2.6 Adding model tests .................................................... 11
      2.6.1 Setting muscle parameters ..................................... 13
   2.7 Viewing test results ................................................. 14

3 Muscle model 17

References 18
1 Introduction

The purpose of this software is to guide the user, through a Graphics User Interface (GUI), to estimate muscle forces using an EMG-driven model. It basically automates the processes that has been used by our group to process EMG signals and input them into a dynamic model of skeletal muscles, which is numerically integrated. This process gathers several steps, which we suggest to be followed towards a 'correct' estimate of muscle force.

EMG-driven model muscle force estimation is an active area of research by several groups throughout the World. Each of these groups have been developing their own analytical, experimental and numerical approaches, more or less similar among each other. In this software, we have followed the procedures, models, filtering etc. that are used in our own research. In any case, EMG-driven model muscle force estimation still presents several open questions (see, for example, [9] and [1]). For example, our current approach avoids using parameter optimization, to avoid masking eventually large modeling errors behind unrealistic parameter adjustments, as pointed out by [5].

Our intention is to periodically release new versions of the software, as new features and improvements are incorporated in the muscle dynamic model and in the EMG collection and processing procedures. One major limitation of the current version is that only isometric analysis can be performed, which we hope to overcome in the next release.

In the current version of the software, a particular formulation of muscle dynamics has been implemented [8], that has been extensively used by our group in several works. The user can implement his/her own model by changing the files Model.m and mmusc4.m. Please see Sec. 3.

The functional unity over which the software is conceived to operate the isolated joint. It is possible to infer the accuracy of the EMG force estimation by comparing the total joint torque with the sum of the individual muscle forces multiplied by their respective moment arms. Joint torque, by its turn, is expected to be measured by a dynamometer. Some authors uses inverse dynamics to estimate joint torques. Such approach allows a much greater flexibility in the choice of the biomechanical situation to be analysed. However, it presents a series of drawbacks, that must be considered consciously: impossibility to analyse isometric tasks (no kinematics at all), difficulties assess the torque error level, necessity to measure kinematics, multibody modeling inaccuracies etc.

The current distribution was conceived for analysing lower-limb muscles, which literature parameters are provided as defaults. However, other muscles can be tested as well using 'Generic Muscles (1-6)', but no default parameters are provided.
1.1 Signal collecting recommendations

Before collecting the EMG signals, please note the following recommendations:

- If you do want to analyze the torque produced by the muscle against an independent measurement of torque, a torque curve provided by a dynamometer (e.g. CYBEX or BIODEX) and synchronized with the EMG must be provided;

- Otherwise, it is possible to provide a *dummy* column of data that will be assigned as the torque, and the corresponding comparative analysis between the two torque curves will obviously meaningless;

- Install the EMG electrodes in the subject in the appropriate positions for the muscles you want to analyze. We strongly suggest following the SENIAN recommendations\(^1\);

- Collect at least two series of Maximum Voluntary Contraction (MVC) trials, with about 10 seconds of duration;

- Don’t remove the electrodes nor change any data acquisition parameters between the MVC trials and the actual EMG recording;

- It is necessary having the time stamps uniformly distributed for each sample in one column of the data file\(^2\);

- IMPORTANT: Start acquiring the signals, with the muscles as relaxed as possible, by some seconds before starting the motor task. This patch of signal will be used to find the offset base line of the EMG signal.

2 Usage of the program

2.1 Running the program

To execute EMG Driven Model Force Estimator (EMGD-FE), you must first have MATLAB installed on your computer. The program has been designed and tested for MATLAB R2009, and requires Signal Processing toolbox to run.

\(^1\)http://www.seniam.org/

\(^2\)If the time vector is not completely uniform due to hardware performance limitations, you can generate an uniform time vector and resample the EMG or torque data using MATLAB `interp1` function.
After downloading EMGD-FE, extract the downloaded .zip file into a convenient folder. To run the program, open MATLAB, and change your path to the folder where you extracted the program. The program can be then run by entering `Main` into the command window. You should then see the window as in Figure 1.

### 2.2 Choosing a display language

EMGD-FE is distributed in two languages: Brazilian Portuguese and English. To switch between available languages, click the 'Change Language'/'Alterar idioma' button and select the desired language from the menu.

### 2.3 Creating a new test subject

Before adding the EMG signals and applying the model, you must first create a file, that will contain both the information on the test subject and its EMG data. This information is stored in a `.emg` file, which is a MATLAB `struct` file.

If you have already created a `.emg` file in a previous session, you can load it by clicking the 'Open' button in the top left corner and selecting the file.

Otherwise, you must create a new subject, by clicking the 'New' button, and then selecting the location where the file will be created. Then, insert the name of the new test subject, its age (optional)
and a comment (also optional). To edit any of these pieces of information, click the 'Edit' button.

We have include a sample subject (named BSG), which files are saved in the 'sample_quadriiceps' subdirectory. A sampling frequency of 2 Hz was used, analogically pass-band filtered from 35-500 Hz. The following files are included:

Table 1: Sample files characteristics.

<table>
<thead>
<tr>
<th>file</th>
<th>content</th>
<th>data</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSGMVC1.txt</td>
<td>1st MVC</td>
<td>time (s), torque (Nm), EMG VM (mV),</td>
<td>10 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMG RF (mV), EMG VL (mV)</td>
<td></td>
</tr>
<tr>
<td>BSGMVC2.txt</td>
<td>2nd MVC</td>
<td>time (s), torque (Nm), EMG VM (mV),</td>
<td>10 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMG RF (mV), EMG VL (mV)</td>
<td></td>
</tr>
<tr>
<td>BSGL.txt</td>
<td>vastus lateralis EMG</td>
<td>time(s), EMG VL (mV)</td>
<td>50 s</td>
</tr>
<tr>
<td>BSGM.txt</td>
<td>vastus medialis EMG</td>
<td>time(s), EMG VM (mV)</td>
<td>50 s</td>
</tr>
<tr>
<td>BSGR.txt</td>
<td>rectus femoris EMG</td>
<td>time(s), EMG RF (mV)</td>
<td>50 s</td>
</tr>
<tr>
<td>BSGT.txt</td>
<td>torque</td>
<td>time(s), knee extension torque (Nm)</td>
<td>50 s</td>
</tr>
</tbody>
</table>

The particular way that the data is distributed among the files has been arbitrary chosen for this example. You can arrange your data file as you wish, and choose latter the correspondence between the columns and the collected signals. Evaluating the torque estimated by the EMG-driven model requires both MVC (maximum voluntary contraction) and submaximal tests for the subject. Another set of data for one triceps surae (gastrocnemius lateralis, gastrocnemius medialis, soleus) test was also included (sample_triceps directory)^3.

2.4 Adding an MVC test

On the 'Show' tab, select 'MVC tests', then click the 'Add' button, to open a dialog. A new window should open.

Give a name to the test to identify it, and, in the 'Data' tab, click the 'Add' button to select which muscles correspond to this test. Only lower-limb muscle names (and their corresponding default parameters) are provided. If the user wants to simulate other muscles, please use the Generic Muscles (1-6). For each of the muscles, click the 'File' button to select the text file containing the corresponding data. This file should contain the test data disposed in columns, as in Figure 3.

---

^3 In this case, the MVC test data is presented in separated files for each muscle and torque.
EMGD-FE will try to guess the sampling rate based on the first column of the data file, which is supposed to be time. It verifies if the 10 first samples are linearly spaced. Otherwise, please provide the sampling rate in the corresponding box.

After selecting the file, inform which column contains the data with the corresponding signals, either torque or EMG (see the drop-down list in lower right part of the screen to change the current column. If all the torque is in the same file of the EMG data, you can check the 'Same file as Torque'\(^4\). Please repeat the operation until all desired muscles have been included in the analysis.

When you have finished adding muscle data, press 'Ok' to enter into the filter window.

### 2.4.1 Filtering test data

After the Torque Filter window appears, you have the option to apply a low-pass filter in the torque data\(^5\). The filtered signal can be previewed alongside the raw data by selecting the 'Show data' box, and inspected further by clicking 'View large'. When you finish filtering the torque data, press 'next'.

After that, you are given the option to remove signal offset. This is done by selecting a flat region corresponding to almost-zero torque, considering that the muscle is completely relaxed. Use the mouse or type the desired offset time interval in the textbox. To undo the offset removal, click 'Reset'.

In the next window, select the filter parameters for the EMG signals and press 'next'. Three filters

---

\(^4\)If you are not measuring torque by an external device, such as a dynamometer or estimating it through inverse dynamics, please provide a dummy torque column with any desired values.

\(^5\)A Butterworth forward and backward IIR digital filter is used.
are applied:

1. The first is band-pass filter to remove artifacts. The second is used to remove mains hum (electrical network background noise and its 3 harmonics)⁶. Provide the network frequency which is used in your country. The third extracts the EMG signal envelope using a low-pass filter. Usually, a very low cut-off frequency is used.

In the next window, select the MVC epoch which you consider as being maximal muscle activation for further EMG normalization. Commonly, a 2 to 3 seconds epoch of maximal sustained torque is chosen for that purpose. Press 'finish' to end the MVC analysis.

2.4.2 MVC analysis

When you have finished entering and processing MVC data, you can choose a particular set of data (torque or filtered MVC EMG) and press 'analyze' button on the right of the screen. If you press 'edit', the filter and offset removing tool is called (2.4.1). By pressing 'export' it is possible to save a .txt file with the raw input and processed data from your MVC analysis. Data columns appear in the saved file in the same order they are selected in the 'export' window. 'Remove' erases the file selected in the list at the left part of the screen.

The screen shown in Figure 4 will appear. Drag the mouse to select a time interval and write down in the 'equation' box the Matlab expression will want to evaluate, e.g. mean.

If you have performed additional MVC tests, you can repeat the procedure pressing 'add' button, choosing other names for the additional MVC tests. All MVC test names you have uploaded are shown at a list in the left part of the main window.

2.5 Adding the main 'Submaximal' test

On the 'Show' tab at the right upper corner of the main screen, select 'Submaximal tests', then click the 'Add' button, to open a dialog. By pressing 'Add' a window similar to that used in the MVC file input will appear. Please choose the files containing the torque and EMG data, selecting for each file the columns accordingly (select the file column number at the box at lower right corner. When you have finished entering the data, press 'next'.

The filtering and offset removal procedures are similar to those performed in the MVC tests (see sec. 2.4.1). Also, it is possible to edit and export data, similarly to MVC analysis (sec. 2.4.2).

⁶The Matlab routine newSIm has been developed by the Laboratorio di Ingegneria del Sistema Neuromuscolare e della Riabilitazione Motoria, Politecnico di Torino, Italy. Used with permission.
Figure 3: MVC editor screen.

Figure 4: MVC analyzer.
If you have performed additional submaximal tests, you can repeat the procedure pressing 'add' button, choosing other names for the additional submaximal tests. All submaximal test names you have uploaded are shown at a list in the left part of the main window.

2.6 Adding model tests

To begin the analysis of the data you have uploaded, on the 'Show' tab, select 'Model tests’, then click the 'Add’ button. The window bellow will appear.

Figure 5: Model test window

Please provide in the upper left corner, at the 'Name of new test’ tab, a name for the model analysis. Chose the 'Submaximal test to model’ the submaximal test you want to analyze. In the 'MVC test to normalize the model’ tab, choose which MVC test you want to consider for normalizing the EMG data.

In the lower part of the window, the muscles which you have provided EMG data are listed on the left. Sometimes, deep muscle EMG signals are not available and can be estimated. See, for example, Oliveira and Menegaldo [11]. Click 'add' button to choose from a list which muscles will have their EMG inputs estimated. Press 'ok’.

In the next screen (see Figure 6), write down the formula you prefer using to estimate the unmeasured EMG. For example, if you want to estimate vastus intermedius activity as the mean between vastus lateralis and vastus medialis, write '(vasmed+vaslat)/2’. When you press 'enter’ key, the new signal is shown. You can try other functions as well, and when you have finished, press 'next’.

A new screen will appear (Figure 7) showing the normalized EMG activity of all muscles, measured and estimated. Drag the mouse to select a time interval which you presume that the muscles are all relaxed, preferably before the motor task begins. The system calculates the mean excitation inside this interval, assuming it as the initial condition for integrating numerically the muscle dynamics. Press 'next’.
Figure 6: Estimating non-measured muscles EMG.

Figure 7: Definition of the initial values for the state variables to integrate the model.
2.6.1 Setting muscle parameters

A screen with default model parameters (Figure 8) will be shown, where you can see and edit muscle dynamics parameters. In the right, the global parameters that apply to all muscles are set. Below such parameters, it is possible to set scale factors to maximum muscle force, optimal fiber length and moment arm [8]. In the right/center, the specific muscle parameters are defined. If you check ‘set maximal force directly’ box, default or user defined values are used. Otherwise, insert your own value of ‘physiological cross-sectional area [cm²]’. Maximal force will be calculated multiplying this by the ‘specific tension [N/cm²]’ defined on the left. This area can be estimated using some medical imaging technique, such as ultrasound [12]. The default parameter values are taken from Delp lower extremity SIMM model [2]. Muscle names are defined in the file lang-en.xml or lang-pt.xml and default parameters values in defaults.xml, which can be edited by the user. If a Generic Muscle is set, 0 value default parameters are set.

Moment arm must be provided by the user, since it varies widely depending on the Degree of Freedom and joint position. We suggest performing a previous study using Opensim [3] or using regression equations [7] to find the most suitable values for the particular body position your EMG signal has been collected. If a muscle is antagonist relative to your positive torque convention, setup the moment arm as a negative value. For the sample test provided together with the software distribution, use 0.048 m for the knee extension quadriceps moment arm.

In the right extreme of the window, a list of free parameters can be defined. The first, called A, corresponds to the ‘A-parameter’ of the activation dynamics non-linearization scheme, proposed by Manal and Buchanan [6]. The greater the value of A, more bulged the neural activation x muscular activation curve becomes (Figure 9). The effect of increasing A is magnify low-activation levels relative to high activation. It is possible having an idea of varying systematically the A-factor in a quadriceps EMG-driven model study [10].

The generic parameters P1, P2,..., P9 can be arbitrarily configured by the user. They are available by the Model.m routine, which calls the m-function where the dynamical model is defined and integrated (see Sec. 3) and Appendix. In the current distribution of this software, the parameters P1, P2 and P3 are used as follows. We ave modeled variable relationships for pennation angle as a function of muscle length only for soleus, gastrocnemius medialis and gastrocnemius lateralis [10], using data from Kawakami et al. [4]. If you prefer using variable pennation angle, only for these muscles, set P1=1. Otherwise, let it empty, or set any other number. These authors have collected triceps surae pennation angle and length data for either the knee extended or flexed, and in high and low activation
levels. Depending on the situation that most fit user’s actual analysis, it is possible to choose \( P_2 \) and \( P_3 \) parameters as follows:

- \( P_2=1 \), knee extended
- \( P_2=2 \), knee flexed
- \( P_3=1 \), high activation level \( (u \approx 1) \)
- \( P_3=2 \), low activation level \( (u \approx 0) \)

In the current version, the additional parameters \((P_4-P_9)\) are not used and can assume any value. After you have setup the parameters for all muscles of your analysis, press ‘finish’ button to run the numerical integration of the EMG-Driven model.

![Figure 8: Muscle model parameters](image)

### 2.7 Viewing test results

When numerical integration is finished, the screen shown in Figure 10 allows the user to perform a series of analysis. By default, the list of possible analysis shown in center of the screen are implemented: error between measured and estimated torque, total estimated torque, individual muscle torques, percentile contribution of each muscle to the total torque and individual normalized EMG signals.

Pressing ‘analyze’, you can apply arbitrary mathematical formulæ to the data, using the mouse to select the time interval, as usual (Figure 11). You can also plot several results in the same matlab figure, by pressing ‘view’. Selecting additional results, every time you press ‘view’ another plot is added to the same figure (Figure 12). You can also remove and export data, as addressed above in sec. 2.4.2.
Figure 9: Neural activation x muscular activation (A-model)

Figure 10: Model tests results
Figure 11: Analyze muscle model integration results.

Figure 12: View results in a matlab figure.
3 Muscle model

A description of the particular formulation of the Hill-type musculotendon actuator can be found in [8]. Essentially, it follows Zajac [14] model, adimensionalization scheme and notation, but includes elastic and damping elements parallel to the contractile one, mainly to improve numerical robustness (Figure 13).

Activation dynamics is a bilinear 1st order differential equation with the formulation presented in [13]. Model formulation is the same for all muscles (with exception of a set of parameters) and comprises three state equations and variables: activation, tendon force and muscle (contractile part) length.

Muscle dynamics model can be modified by the user directly in the matlab functions where it is defined: Model.m and mmusc4.m (see Appendix) In Model.m, you can define additional parameters adding elements to a vector of parameters defined as a global variable, and call them from inside the mmusc4.m function. Observe, by the end of Model.m function, that the dynamical equations written in musc4.m are integrated using a Runge-Kutta algorithm, ode45, by default.

\[ [t,y] = \text{ode}45( '\text{mmusc4}', \ \text{time}, y0 ); \]

Matlab has several integration algorithms, and depending on the characteristic of your differential equations (stability, stiffness etc.) one or other will be more efficient. You can try implementing or modifying the muscle dynamics equation creating your own function with a layout similar to mmusc4.m. Our group uses for some limited muscles, such as triceps surae, an alternative formulation [9] that includes a 'non-linearization' step of the activation dynamics, the so called 'A-model' [6]. This formulation can also include a variable pennation angle relationship (see Sec. 2.6.1). Muscle dynamics an active area
of research in our group, and future versions of this software are expected to progressively include new formulations. However, the user is invited trying his own model changing the `mmusc4.m` and `Model.m` accordingly.

**References**


