**Exercise 1** **November 16th, 2023**

## General Information

This exercise will guide you to various surface EMG data postprocessing and analysis steps. We suggest using MATLAB to visualize, quantify and process the EMG data and corresponding force signal during the exercises. Note that using Python is also possible, although we will use MATLAB for the exemplary solution. If you have not installed MATLAB yet, click on the link below and take advantage of FAU's Campus-Wide License to install and use MATLAB on your computer: www.mathworks.com/products/matlab/student.html

In addition to the base MATLAB version, you might need the MATLAB signal processing toolbox (depends on the functions you want to use). You can install it in MATLAB under Home -> Add-Ons.

MATLAB provides a lot of built-in functions that can make your life easier during these exercises. It is always a good idea to check the **MATLAB documentation**. Don´t hesitate to also use Google or **write in the StudOn forum**. Also check the **Tips** section on the last page.

**Submission:**

1. Your code as a .m (or .py) file (Your main script as well as any potential self-defined functions).
2. A .pdf file with your results. The results that should be included in the report are **written in this colour** (You can save the plots as a .png with *save as* in the plot window). For questions that require written text as an answer, 1-3 sentences or bullet points should usually be enough. Plots need to contain labelled axes and suitable units.

**!**

- Please **do not** submit a document with both the code **and** the plots & answers! Only submissions with two **separate** files, particularly a .pdf with the plots & answers, and a separate .m/.py/.ipynb file containing your code will be accepted.
- This is an individual exercise. If the submitted code and/or .pdf file is identical to another submission, both submissions will be marked as failed.

Submission Deadline: **06.12.2023, 23:55 pm**

Note that the interpretation of the exercise results and the physiological concepts behind these **are relevant for the exam. Additionally, there will be bonus points for submitting both exercises.** Note the bonus points will only be granted if you submitted **adequate results**.

# The Dataset

Along with this document, you received two data files: *Slow_Contraction.mat* and *Rapid_Contractions.mat.* Both data contains surface-EMG data from 64 electrode placed on the tibialis anterior muscle. In the *Slow_Contraction.mat* file*,* the subject performed isometric ankle dorsiflexion trying to follow a predefined force ramp with a plateau force amplitude of 50% of the subjects maximum force (maximum voluntary contraction MVC) (see Fig. 1 for the experimental protocol).

In the *Rapid_Contractions.mat* dataset, the subject performed 5 isometric ballistic contractions followed by 3-second holding of the maximum force. A ballistic contraction is a contraction, that is performed as fast and strong as possible.

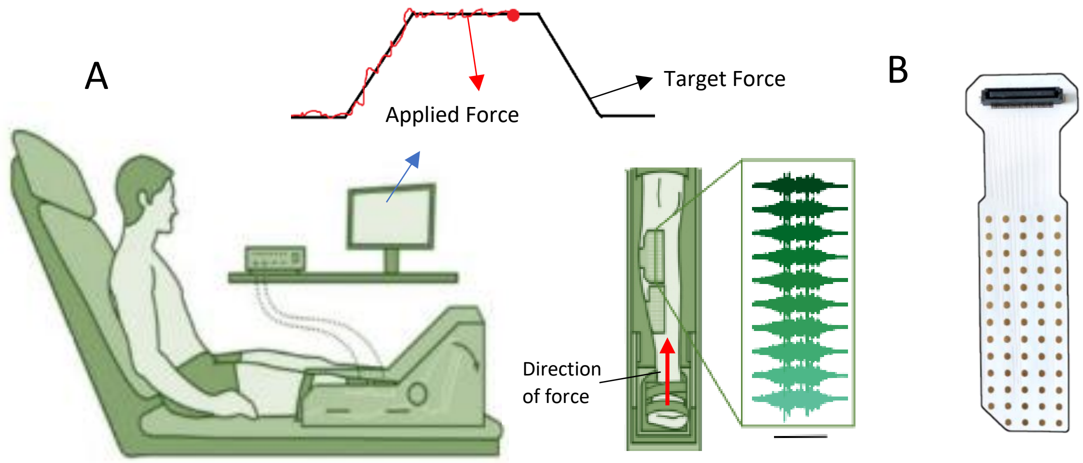


***Figure 1****. **A:** Experimental protocol. Foot fixated in an ankle dynamometer with force transducer measuring ankle dorsiflexion force. 64-channel surface-EMG grid records data from subjects tibialis anterior muscle during isometric contractions. Target force ramp visible on screen for the subject to follow. From: A. Del Vecchio, D. Farina (2020). **B:** 64 channel high-density surface-EMG electrode grid used in hte data acquisition. Electrodes are aranged in a 13 x 5 matrix. Each electrode corresponds to one channel of EMG signal in the resulting dataset.*

Both datasets contain the following variables:

- **EMG signal (`SIG`):**
  The raw EMG signals in mV of all 64 channels stored inside an 13x5 cell matrix, according to each channels location within the 13x5 channel electrode grid (see Fig. 2B)

- **Force signal (`ref_signal`)**
  The force signal measured by the force sensor in mV

- **Sampling frequency (`fsamp`)**
  Sampling frequency of EMG and force signals in Hz

- **Signal length in second (`SIGlength`)**
  Length of EMG and force signal recording in seconds

- **Other variables, not relevant for this exercise**

## Task 1. Data preprocessing & visualisation

Visualizing the signals is always a good idea to get a first impression and feeling for your data. Use the *Slow_Contraction.mat* dataset.

1.1 Convert the force transducer signal to Newtons using the device-specific conversion factor. $Force\ in\ N = \frac{Force}{ConversionFactor} * Gravity$ *ConversionFactor = 0.02; Gravity= 9.81 m/s*$^2$

1.2 **Plot the force signal** in Newtons on the y-axis with the time in seconds on the x-axis. Include axis labels. Take a close look at the signal. **Can you observe anything noticable?**

1.3 The force signal contains high-frequency noise, that will cause problems in later analysis steps. **Filter the force signal with a suitable filter.** Take care to only remove the noise, but to preserve all valuable information in the data. **Plot the unfiltered and filtered signal in the same plot, so the difference between the signals is visible** (choose suitable visualisation window).

1.4 **Plot** an exemplary channel of the EMG data together with the force signal (same or different plot). **How do force and EMG signal relate**?

## Task 2. Force steadiness

Calculating the force steadiness during the plateau phase is a basic first analysis of the force data. Use the *Slow_Contraction.mat* dataset.

2.1. Compute the **coefficient of variation (CV)** during the plateau phase of the force. CV can be calculated by dividing the signals standard deviation by the signals mean, multiplied by 100 to get the value in %. $CV = \frac{Standard\ Deviation\ \sigma}{mean\ \bar{x}} * 100$

2.2. Think about the meaning of this analysis. **What does it mean physiologically?**

## Task 3. Rate of force development

Here you compute the rate of force development for the ballistic contractions dataset (*Rapid_Contractions.mat).*

3.1 Pre-process and **visualize** the force signal as in Task 1.

3.2 Compute the Rate-of-Force Development (RFD) of the first 500 ms of each contraction in timesteps of 1 ms (or 2 samples, as the sampling frequency is 2048). Search how to calculate the RFD in the literature [1][2]. You can define the onset of each contraction either visually or automatically. **Visualize the values in a suitable way.**

3.3 **What is the physiological relevance? In what demographics/populations could the RFD be important?**

## Task 4. Root-Mean-Square (RMS) of EMG

The RMS is an important feature in EMG signal analysis that provides insight into the amplitude of the physiological activity by giving a measure of the signal power. It is calculated as the square root of the signal mean in a moving window. As you compute the square root, this also rectifies the signal. Perform the following steps for **both datasets**.

4.1. Convert the cell array *SIG* into a 2D array of shape *channels x samples* and compute the average across all 64 channels.

4.2. Compute the RMS of this average as a moving average with a window length of 200 ms. Perform this for the EMG data of both datasets. **Visualize the result.**

4.3. Visualize the correlation between the resulting RMS and the respective force signal by **plotting the normalized RMS (x-axis) against the normalized Force (y-axis)** in the form of a scatter plot. Then quantify the correlation by computing and reporting the **correlation coefficient R**. Interpret the result and **explain the physiological meaning**.

## Literature

Access the literature via the university VPN.

1. Comfort P, Allen M, Graham-Smith P. Comparisons of peak ground reaction force and rate of force development during variations of the power clean. J Strength Cond Res. 2011 May;25(5):1235-9. doi: 10.1519/JSC.0b013e3181d6dc0d.
2. Haff GG, Ruben RP, Lider J, Twine C, Cormie P. A comparison of methods for determining the rate of force development during isometric midthigh clean pulls. J Strength Cond Res. 2015 Feb;29(2):386-95. doi: 10.1519/JSC.0000000000000705.

# Tips

- **Script Structure:** To keep your script well-structured and clear, you can create script sections by writing `%%`, followed by a blank space and a section title. You can execute single script sections with the `Run Section` button in the toolbar or `Ctrl+Enter`, instead of having to execute the entire script. (https://www.mathworks.com/help/matlab/matlab_prog/create-and-run-sections.html)

- **Data Storage:** When loading the dataset into the workspace using `datasetX = load(filepath)`, the variables are combined into a `struct`ure `array (struct)`, which is a very convenient way of storing variables that belong to the same dataset, as they can be easily accessed via dot notation as in e.q.: `datasetX.SIG{row,col}`. This keeps the workspace clean when working with multiple datasets, and also allows to load datasets that contain variables with the same name, as those variables would overwrite each other otherwise. You can add more fields to the `struct` by using the same dot notation, e.g. if you want to add the a new data variable *SIGfilt*: `datasetX.SIGfilt= yourFilterfunction(datasetX.SIG,…)` (https://www.mathworks.com/help/matlab/ref/struct.html)

- **Custom Functions:** If you need to apply the same data processing steps to multiple sets of data, is it advised to write a custom function for these steps, that you can then pass the datasets to. A function can either be saved in a separate .m file or defined at the bottom of your main script. Check https://www.mathworks.com/help/matlab/ref/function.html on how to define functions.

- **Data Plotting:** If you want to plot multiple related figures into one figure window, check out the `subplot` command. If you want to plot multiple plots into the same figure, use the `hold on` command. `yyaxis right` lets you plot multiple data series into one plot with independent y axes.
(https://www.mathworks.com/help/matlab/ref/subplot.html)
(https://www.mathworks.com/help/matlab/ref/hold.html)
(https://www.mathworks.com/help/matlab/ref/yyaxis.html)
**Important**: Don´t forget to label your axes in your plots and to give them informative titles.

- If you do not know how to code a certain task, there is probably a **built-in MATLAB function** for you. Check the documentation, Google or the AI language model of your choice.