# **Auto C3D Checker - User Guide**

## **What is this?**

This is a tool for automatically evaluating the quality of a biomechanical trial (in .c3d format) involving the use of marker-driven motion capture, forceplates (from here on referred to as FP), and surface EMG.

This tool evaluates the .c3d in terms of its appropriateness for neuromusculoskeletal (from here on referred to as NMS) modelling. Specifically, this tool automatically determines:

* Whether Markers crucial for tracking are present throughout the trial
  + If yes, what is the motion direction throughout the trial
* If the instrumented leg hits a FP during the trial
  + If yes, which FP is hit and does the foot/both feet hit this FP fully or only partially
  + If FP is hit, at what frame do foot strike and foot off happen
  + If all above are found, the required trial length (foot strike to foot off, plus extra frames for analysis window and EMG delay) is calculated
* If the recorded trial is long enough for the required trial length
  + If not, is there sufficient data to account for EMG delay while padding the trial
  + If yes does start/end/both need padding
  + If required, padding is then performed
* Whether EMG signal quality is sufficient for EMG-assisted NMS modelling
  + If not, how many of the total EMG in the trial are of insufficient quality

Based on these .c3d quality criteria, trials are modified (i.e., start padding) and divided into three groups:

* Calibration – Highest quality trials (where the instrumented leg fully hits a FP and trials have required length) without a need for any data modifications and with at least 75% of EMG being of sufficient quality. These trials can be used to calibrate the NMS model.
* Execution – Trials (where the instrumented leg fully hits a FP) with a need for data modifications (i.e., padding) or with insufficient EMG quality. These trials can be used to run the NMS model, but not to calibrate it.
* Unusable – Trials where the instrumented leg does not hit the FP (or not fully), or where not enough data is present to account for EMG delay for padding.

The checker outputs a “Results.xlsx” containing the results of all the abovementioned checks for all participants. Further, it creates an “EMG Figures” folder for each participant, where the classified EMG can be found. Furthermore, it creates an “InputData” folder and places the modified Calibration/Execution .c3d trials into appropriate folders.

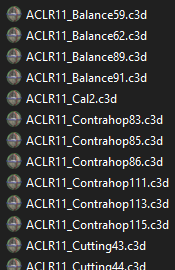
## **Limitations**

Foremost, consider the following limitations to know if this code is of use to you:

* **Multiple foot strikes per FP** – The script finds a single foot strike per FP, meaning if you have multiple foot strikes for the same FP (which is non-standard in gait analysis), only the first one will be analysed
  + E.g., treadmill trials are not suitable for analysis with this script
* **Channel/Marker names starting with numbers instead of letters** – Variable names in Matlab cannot start with a number and are automatically renamed by btk (e.g., EMG name in .c3d 1BFLH is renamed to **C**1BFLH), thus it’s necessary to rename such EMG channels with the included overwriteEMGNames function (set overwriteEMGNames = true; in the script and provide original & new names in the autoC3Dsetup.xml under the EMGs tag.
* **Channel/Marker names having a dot** – Variable names in Matlab cannot include a dot and are automatically renamed by btk (e.g., FP force name in .c3d Force.Fx1 is renamed to Force**\_**Fx1), thus it’s necessary to rename those with the included overwriteFPNames function (set overwriteFPNames = true; in the script and provide original & new names in the autoC3Dsetup.xml under the FPs tag.
* **Simultaneous foot strike on two FP** – If a foot simultaneously strikes two FP, the script will designate this event as ‘full foot on FP - false’, which is only an issue if the two FP are touching (no distance between corners) and thus could be combined to calculate a single GRF
* **Inconsistent number of markers/EMG/FP between trials** – The script expects a consistent number of markers/EMG/FP between all investigated trials and will not work correctly if the number of e.g., FP varies. For such cases, separate the trials into consistent parts and make a autoC3Dsetup.xml for each and run the script for every part
* **Foot on FP for static trials** – The script will not detect foot on FP for trials where person stands on FP for the whole duration of the trial (e.g., calibration/static trials), but this information is not relevant for standard NMS modelling
* **Trials without recorded EMG** – The script can be run on trials without EMG by setting runEMGClass = false; in the main script, but this is not standard for NMS and thus has not been tested extensively

## **Inputs**

To use the Auto C3D Checker, the following folder structure is required (see an example in the included Sample Data):



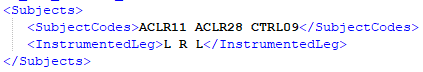


The “Base Folder” contains participant folders, which contain the .c3d files (it’s best to have the participant in the .c3d files name to avoid confusion).

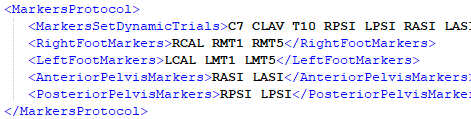
Furthermore, an .xml setup file providing additional information and settings is required (autoC3Dsetup\_example.xml can be found in the “templatesXML” folder):



CoordinateSystemOrientation – Orientation of the lab coordinate system needed for automatic trial motion direction detection. First axis is the motion direction throughout the majority of your trials, second is the vertical axis, third is the one that remains (for further information see: https://rehabenggroup.github.io/MOtoNMS/manual/acquisitionInterface.html#setup-files)

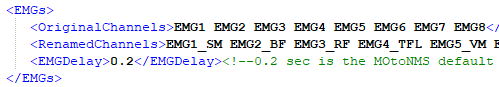


SubjectCodes – Participant folder names in your Base Folder.  
  
InstrumentedLeg – Instrumented leg for the participants (same order as the participant folders)

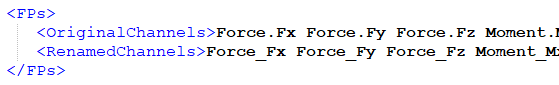


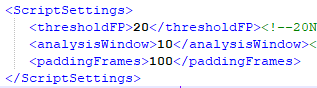
MarkersSetDynamicTrials – Required markers in all dynamic trials (if required markers are missing, script might throw an error)   
  
Right/LeftFootMarkers – Markers on the right/left foot (used to determine foot strike and full foot on FP)

Anterior/PosteriorPelvisMarkers – Markers on the pelvis/hip, two on the front of the pelvis/hip and two on the back, used for motion direction detection   
***Note:*** **If** **no PSIs** markers are available, **choose a single back marker at midline of the body** (L1-L5)



OriginalChannels – Channel names of your EMG in all .c3d  
  
RenamedChannels – Optional entry to rename EMG channel names in all .c3d (this option is enabled in the script with overwriteEMGNames = true;) disabled in the script if not required   
  
EMGDelay – Value for electromechanical + hardware delay (0.2 sec is the MOtoNMS default, don’t change without a good reason!)

  
OriginalChannels – Channel names of your FP in all .c3d without the FP number  
  
RenamedChannels – Optional entry to rename FP channel names in all .c3d (this option is enabled in the script with overwriteFPNames = true;) disabled in the script if not required



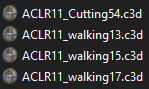
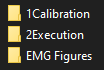
thresholdFP – FP threshold, only change if encountering errors  
  
analysisWindow – Extra frames for analysis window (10-25 is standard for CEINMS)  
  
paddingFrames – How many frames to pad trials with (100 is sufficient in most cases)



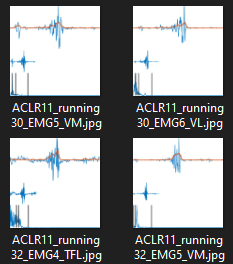
Lastly, the paths to both the base data folder and the .xml setup file need to be adjusted to point at your data inside the main script.

## **Outputs**

1) The script outputs the modified trials for each participant in their respective InputData folders:

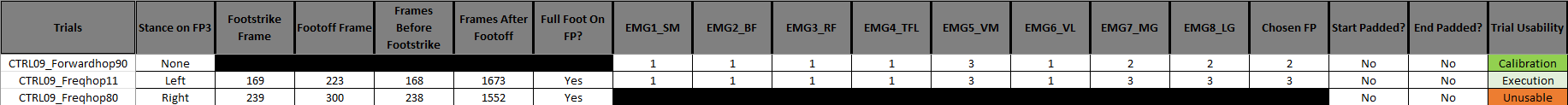


***Note:*** **All trials** in the InputData folder have been adjusted to **start at frame #1** for simplicity of calculations.

2) The EMG are classified and save into their respective class folders:

3) The .c3d quality checking results are saved for each investigated trial (even the unused ones) in an Excel file:





4) A log file containing the Matlab command window output will be saved in the same folder as the “Base Folder”:





## **Bonus Scripts**

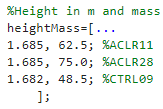
This tool also includes bonus scripts to further automate the NMS pipeline:

### ***autoAcquisitionXML***

This bonus script automatically creates an for each session in your InputData folder (which was created by the main Auto C3D Checker script).

This “acquisition.xml” is based on a template (“autoAcquisitionTemplate.xml”, found in \templatesXML) and is required for further processing by MOtoNMS.

To run this script, you’ll need to:

* Run the main script successfully (getting the “Results.xlsx”)
* Adjust the “autoAcquisitionTemplate.xml” to fit your data
* Adjust the heightMass variable inside the script (it should be in the same order as the participants in the SubjectCodes tag inside autoC3Dsetup.xml) with your participant information
* Sort the 1Calibration and 2Execution folders for each participant:
  + Any decent dataset will have more calibration quality trials than are necessary for CEINMS calibration. Thus, keep 2-4 trials (use walking/running/cutting trials, other types have not been tested in terms of physiological plausibility) in addition to “static.c3d” in 1Calibration and put the rest into 2Execution. One should never execute on the same trials that were used for calibration, so divide the trials sensibly.
  + Repeat this for every participant.

### ***autoElaborationXML***

This bonus script automatically creates an  for each session in your ElaboratedData folder (which was created by the C3D2MAT function, which is part of MOtoNMS, this function may be adapted and included as part of this tool).

This “elaboration.xml” is based on a template (“autoElaborationTemplate \_L/R.xml”, found in \templatesXML) and is required for data processing by MOtoNMS.

To run this script, you’ll need to:

* Run the main script successfully (getting the “Results.xlsx”)
* Run autoAcquisitionXML successfully
* Run C3D2MAT (part of MOtoNMS) successfully
* Adjust the elaboration template with your data acquisition information
* Adjust the filter cutoff frequencies inside the script if desired (6-18Hz is standard)

## **Future direction**

Future direction for this tool includes:

* Custom C3D2MAT and MVC from all trials functions
* Function to re-classify manually re-sorted EMG images (in case you disagree with the automatic classification) and update the Results.xlsx
* Function to automatically generate trial specific excitationGenerator.xml and other CEINMS setup files where bad quality EMG are synthesized (instead of the standard approach of having a single setup file for the whole acquisition)
* GUI to increase the ease of use? (May be scrapped as it’s already easy to use)