Title: CAREN2OpenSim: CAREN to OpenSim pipeline to develop and analyze neuro-musculoskeletal model of human

Authors:

Hossein Mokhtarzadeh

Ashwin Ho

Initial Date: 29 March 2019,

Version Date: 7 September 2022 (v8)

Other contributors:

Song Li, Henry Patrick John Winter, Yingcheng Zan and all the great interns who contributed.

Contact:

Dr. Hossein Mokhtarzadeh

Sr. Research Fellow, The University of Melbourne (UoM)

Email: mhossein@unimelb.edu.au

and Mokhtarzadeh.hossein@gmail.com

Summary:

This toolbox enables CAREN (Computer Assisted Rehabilitation Environment) users to collect C3D files and export them into OpenSim easily. Then the OpenSim workflow is provided in MATLAB via OpenSim API to facilitate the calculation of joint and muscle functions as biomechanical variables. Though we provide this as a CAREN toolbox, the process can be used in any other motion lab or Motek Medical devices such as GRAIL, or MGAIT where C3D files in full-body Human Body Model (HBM) are available.

Introduction:

This report presents how to use collected data (C3D files) from any motion capture (MoCap) lab particularly CAREN lab (Motek Medical BV, Netherlands), bring it in MATLAB using OpenSim API, and then use (trc and mot) or (marker) data to scale the generic model, apply inverse kinematics and inverse dynamic processes. The optimization methods (e.g. Static Optimization or CMC) were used for predicting muscle forces/activations (as suggested by OpenSim) while validating each step. The toolkit aims to simplify the process so that the users only make minor changes to use the whole process in their lab. It should be noted that we use this based on CAREN lab data, however, this process can be used in any lab as mentioned above with some changes to the scripts. The toolbox assumes the full body human data is collected in the lab as C3D files (HBM markerset).

Find this link below for more details: Link

Brief intro to the process and details are below it.

Processes	**Details**
Data cleaning and ready for OpenSim (very important)	This is a very iterative process. Before even we enter data in Matlab, we need to make sure C3D and other relevant files are ready and clean for further processing (NaNs, missing data, Files name, …)
C3D to OpenSim to create .trc and .mot files	Using c3dExport.m (Link), we export all C3D files into OpenSim 4.3.
Visualize the experimental data	Using OpenSim GUI, we can visualize our data. This can be done via Matlab as well. Take note that GRF and Markers will have a shift for visualization only, but the data has been correctly imported.
Scaling the generic Model in OpenSim	Scaling the OpenSim generic model needs proper marker-set definition, weights, etc. This step uses static pose data. See this link.
Inverse Kinematics	Dynamics TRC files will be used to do the inverse kinematics, I have finalized a code to do it in batch format. See details here Link
Inverse Dynamics	Inverse dynamics provides generalized joint moments. So, the methods expanded here on OpenSim API will process all files.
Static Optimization	We used the following process using OpenSim API in MATLAB to assess muscle forces. Link
Next step and especially validation: Is My Model Good Enough?	We need to make sure our models are good enough. So how can we do it? Use this guideline below: (Hicks, Uchida, Seth, Rajagopal, & Delp, 2015).

Table 1: Table outlining all the details of the processes used in the OpenSim analyzation from C3D, DFLOW to Muscle forces.

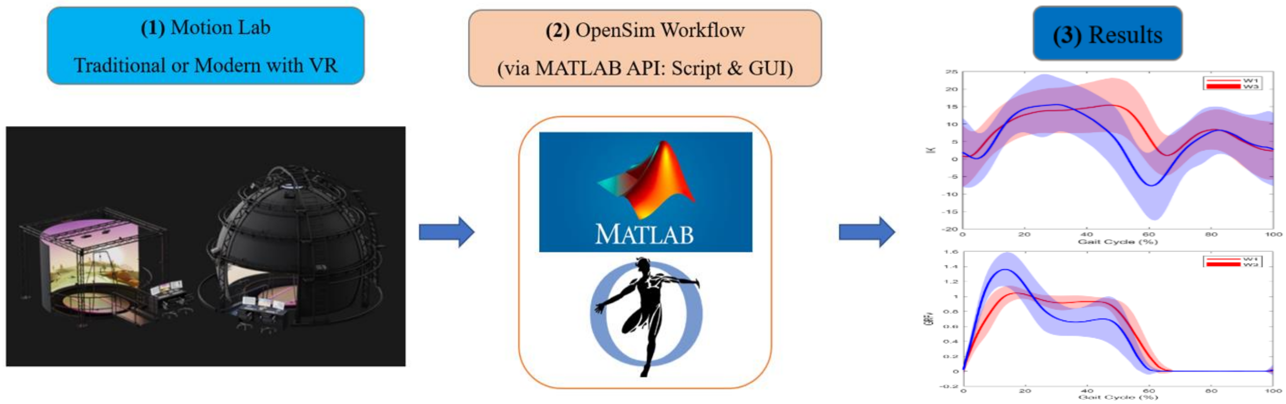


Figure 2: Simplified pipeline of the data processing process using the Caren2OpenSim toolbox.

The pipeline for the toolbox involves the initial gathering of the 3D data with CAREN Lab. The C3D data output file will then go through the OpenSim Workflow via Matlab and can take two paths: Scripting (executing original code) or through a GUI (simplified app). The output of both processes will be plots of the results of the desired biomechanical variables. More information regarding the entire process is described in the Appendix.

A quick setup to use CAREN2OpenSim toolkit in Matlab:

When C3D files are collected in the lab and cleaned up in e.g., Nexus, then the following simple steps will generate all the required outcomes from the OpenSim workflow (scaling-IK-ID-RRA-SO-CMC, etc.). We have tested this toolbox with OpenSim v.4.3, however, the very recent version v.4.4 (July 2022) should be fine with the process as well.

Steps to run the toolbox:

1. Download the package from GitHub: Link
2. Download and install MATLAB and relevant toolbox ('*DSP System Toolbox*' and '*Signal Processing Toolbox*') on your computer if you do not have it installed. We have tested the toolbox in version R2022a
3. Install OpenSim API in MATLAB by following these steps: Link
    a. Copy the mFiles located in (Link) into this directory on your device (C:/Users/<username>/Documents/OpenSim/4.x/Code/Matlab)
    b. Download the OpenSim_requirements zip into your device.
4. Open the following file *c3d2OpenSim.m* and make the following:
    a. In Section 0 (Ln61) of *c3d2OpenSim.m*, named '**Changing paths and destinations**':
        i. add the path to your mFiles (same directory as the files in Step 3a)
        ii. change the destination to your data folder
        iii. add a path to the OpenSim requirement folders
    b. Create a folder called (C3DFiles) then make a copy of all the C3D files into it
5. Then you can easily run the rest of c3d2OpenSim.m and get all the results, as we have provided some sample C3D files here.

Appendix

A detailed explanation of OpenSim Pipeline:

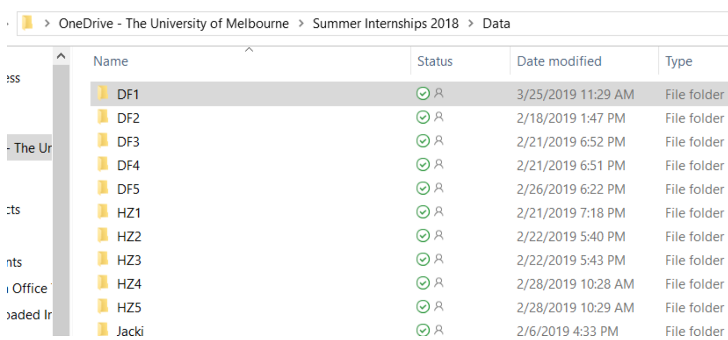
Step 00: Data cleaning and ready for OpenSim (crucial step):

Make sure you check all the C3D files as follows:

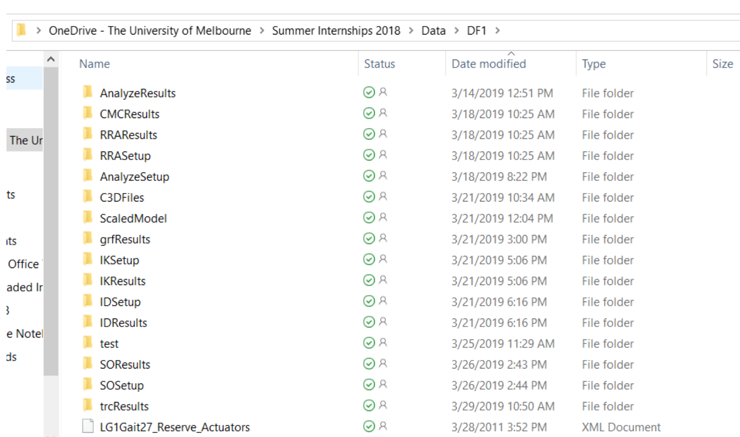
1. Double check the labelled markers and remove or modify missing/incorrect ones. Check them in Mokka or Nexus Vicon to determine if they are labelled correctly with the correct names and trials.
2. If you change the names, remember to match them to the trial.
3. Next, check that all folders are in the respective directory (DFLOW, Vicon, etc.).
4. [next steps]D-FLOW (Usually txt files) - 1 or more files; however, the results or trials might be all together. Vicon may record the trials in different files (.C3D format), which have to be fixed before entering Matlab/Python. *Note: this part will be added in the next versions.*

Step 0: folders and how to input C3D files and other relevant files:

Make sure all files are in one folder, as the codes here are designed for it. For instance, the following shows folders where all the data will be stored:



Each folder has data from CAREN lab which are mainly processed C3D's. Within these folders, create the following folders by running the runOpenSim.m script, or manually create them.



The runOpenSim.m code creates these files in a specific folder. This code also runs the code, c3dExport.m, that writes .trc files into the 'trcResults' folder while writing .mot files into the 'grfResults' folder. All steps following this will read the files from these folders.

All the setup files are in a specific folder while results are written in another folder. Study them to be more familiar and know where to read things if your analysis needs it in the future.

Note that all the C3D files are in a specific folder. This folder must be created before running runOpenSim.m, and copy all your experimental data into this folder.

Then create other files with relevant xml files and select them where necessary (see the steps below) to finalize all the steps from C3D to SO and beyond.

- **C3D to OpenSim to create .trc and .mot files**

Using c3dExport.m (original code), a c3dExportLoop.m was developed where there is an ability to filter the (mot and trc files) and make other changes (e.g. make zero all negative vertical forces).

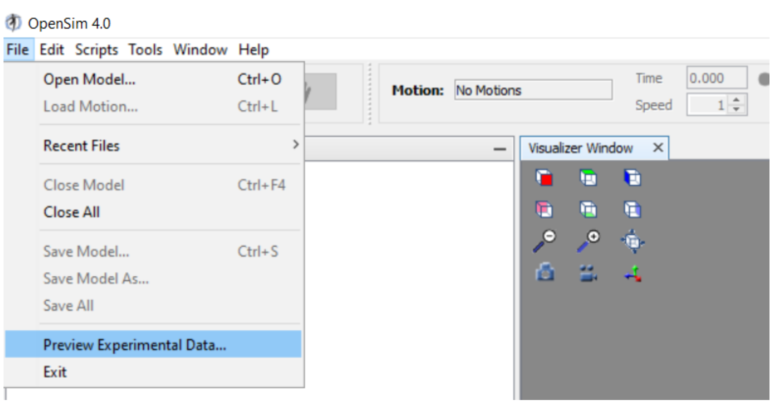
Using runOpenSim.m, we can also change filtering, cutoff and order values, and directory location for files for the process. Make sure all the m files are accessible and can be identified and read by Matlab, and the code will create all trc and mot files in their predefined folders (if they are not there, make them):

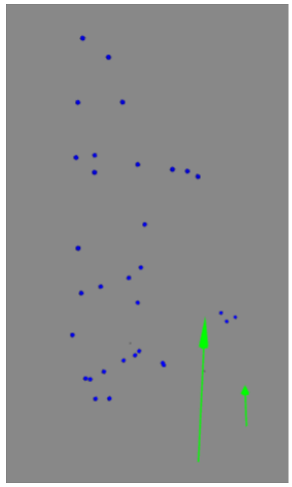
These are the folders following exportation: grfResults, and trcResults.

Note: Event Detection (online or offline), Filtration, cropping the C3D files and/or any other editing are the critical processes that we need to do before:

- **Visualize the experimental data and verify**

There are ways to visualize the .trc and .mot file to make sure we have the right data imported. One way is using the OpenSim GUI, see below:





Another way is to use MATLAB OpenSim API visualizer. ***Tricky, but possible. Currently the "Matlab OpenSim Visualizer" is not working. Manually use the GUI for now.***

Another critical variable that needs to be checked would be CoP (ground_force_1_px, ground_force_1_py, ground_force_1_pz) especially during perturbation. Filtration may change this significantly, so plot them in Matlab or OpenSim to see if they look reasonable. The vertical direction should be fine and must be zero.

- **Scaling the generic Model in OpenSim**

This is a critical and tricky part of the process. Take your time to ensure it is as accurate as possible or it will affect the processes downstream in the workflow.

Based on this link from OpenSim team, there are several steps to consider including, taking a photo of the subject, where the markers are, proper virtual markers definition, joint centre accuracy, etc.

- **Inverse Kinematics (IK)**

With an accurately scaled model, IK and ID processes are straightforward. However, SO might be still quite challenging. IK uses optimization to minimize the difference between experimental markers and virtual ones on the model so the final results show RMS values for each marker and the overall need to follow certain values. Check (Hicks et al., 2015). Below is an example of IK results from a scaled model. Check the out.log.

If we trust our IK data and we are ok with the out.log and RMS errors, then we can move on to ID to find joint moments and forces (generalized ones).

Check the overall patterns using these simple codes below (OpenSim v4.3):

```
% plotting IK plots
import org.opensim.modeling.*

timeSeriesTable = TimeSeriesTable('S+T01_ik.mot');
Forces = osimTableToStruct(timeSeriesTable);
f = fieldnames(Forces);

%plot all joint motions
for i = 1:length(f)-1
hold on
  plot(Forces.(f{i}))
end

figure(1) % plot left and right knee angles
hold on
plot(Forces.(f{10}));plot(Forces.(f{17}))
legend(regexprep(f{10}, '_', ' '),regexprep(f{17}, '_', ' '));
title('Left vs Right knee angle')

figure(2) % plot left and right hip flexion angles
hold on
plot(Forces.(f{7}));plot(Forces.(f{14}))
legend(regexprep(f{7}, '_', ' '),regexprep(f{14}, '_', ' '));
title('Left vs Right hip flexion angle')
```

- **Inverse Dynamics**

ID calculates joint moments, forces (using GRF), and joint motions using the equations of motion (check OpenSim website for more details). Usually, this step should be fast. Sometimes, people do RRA before SO at this stage and adjust their model. However, we can stick to Scaling, then perform IK and then ID and SO for muscle activations.

Note: If this analysis takes too long, then there should be something wrong. For instance, revisit IK and visualize it, GRF and how they are acting and the overall model in terms of joint motions with respect to each other which is good enough for IK outcomes.

I have modified this code: *setupAndRunIDBatchExample.m* for our purpose. Like what we have above on IK, we need to define setup files, where IK results, GRF results and scaled model are. This is the case for the SO process as well. Make sure you point to the right directory. If you input the wrong directory for different fields, don't expect to get the right results. Learn this part well and follow the steps well.

Note: If you have any better ideas to make this more accurate let us know.
Here we need to check out reserve moments and forces. If they are quite large, we need to reconsider things and/or justify them. Usually, if we deal with only the lower limb and do not have a proper upper body model, we will encounter larger reserve moments and forces in this stage and even in SO results.

Note for all: let's find a proper paper or bunch of papers that report joint moments during daily activities esp. walking so that we can refer to and compare our outcomes with them. This is critical for us to move on. If our results are way out of the range reported in the literature, it does not make sense to move to SO.

- **Static Optimization (SO)**

If we are confident in our ID results, we will use SO to predict muscle forces and activation at each frame. This optimization minimizes a cost function, usually the sum of the square of muscle forces or activations (refer to OpenSim website for details).

This is one of the tricky parts of our analyses. We have to make sure we understand every step here.

Make sure we apply proper IK, scaled model (even if it is RRA adjusted model), GRFs and even Reserve Actuators as the muscle may or may not be able to provide and produce the joint torque we found in the ID step. Sometimes people increase muscle isometric forces (MIF) up to 2 or 3 times. Check my papers and others in Pandy's group on usually high-intensity motions like running, drop landing, etc. Others use Reserve Actuators to compensate for such issues. For instance, we use this *(LG1Gait27_Reserve_Actuators.xml)*. However, we need to make sure not to apply large torques and forces as it will compromise the outcome of our SO. This is the tricky part that we need to discuss and understand well. Anyhow, from a technical point of view, I have been modifying the codes to do SO in a loop for all our data.

Here is the code that I have created: *setupAndRunSOBatchExample.m*

This code gets the directory for all the data, and if you already created IKresults, grfResults, and SOSetup and SOResults folders, the code will find them and will continue the SO process for all the data. Finally, the results will be written in this folder: SOResults. Each trial will provide 3 files: 1 xml file and 2 STO files. For instance, these are an example:

W304_StaticOptimization_controls.xml

W304_StaticOptimization_activation.sto

W304_StaticOptimization_force.sto

The code will write the original name of the file (C3D file name) in the beginning of each result in SO outcomes.

Note to all: To get an idea if the results are in the right direction and good enough, one needs to get literature results and compare their findings in the same movements for validation.

- **Next step and esp. validation: Is My Model Good Enough?**

As mentioned above, we need to ensure all the steps are correct or as accurate as possible since they depend on many parameters and even our manual work e.g. in scaling. So after considering (Hicks et al., 2015), we need to write the following codes by asking ourselves these questions:

1. Have we exported C3D files to OpenSim properly?
2. Do we need to filter data before importation or during or after?
3. Are our scaled, IK, ID, and SO models good enough?
4. How can we make sure? Here we need to code and automatize our methods to make sure we are doing the right thing.
5. What steps to take for quality control?
6. Any other questions?

NEXT STEPS - FUTURE WORK

There are other steps that can be implemented in this process including calculating joint reaction forces, induced acceleration of GRF decomposition method, CMC, and Forward dynamics. From the D-Flow side of the CAREN lab, we have implemented some features in MATLAB; however, more is needed to provide a comprehensive toolbox. This process uses C3D files, however, Motek devices can provide Mox files and include HBM where muscle function can be quantified. A simple process is explained here.