Vicon-to-OpenSim Inverse Dynamics Tutorial

Xiaowen Chen

Introduction

This tutorial will guide you through the process of using data gathered from Vicon Nexus to perform inverse dynamics (ID) in OpenSim. Through this tutorial, you will learn how to use inverse dynamics function in OpenSim, how to setup and revise model of OpenSim, how to modify .xml files for OpenSim, and how to use MATLAB to perform ID batch processing.

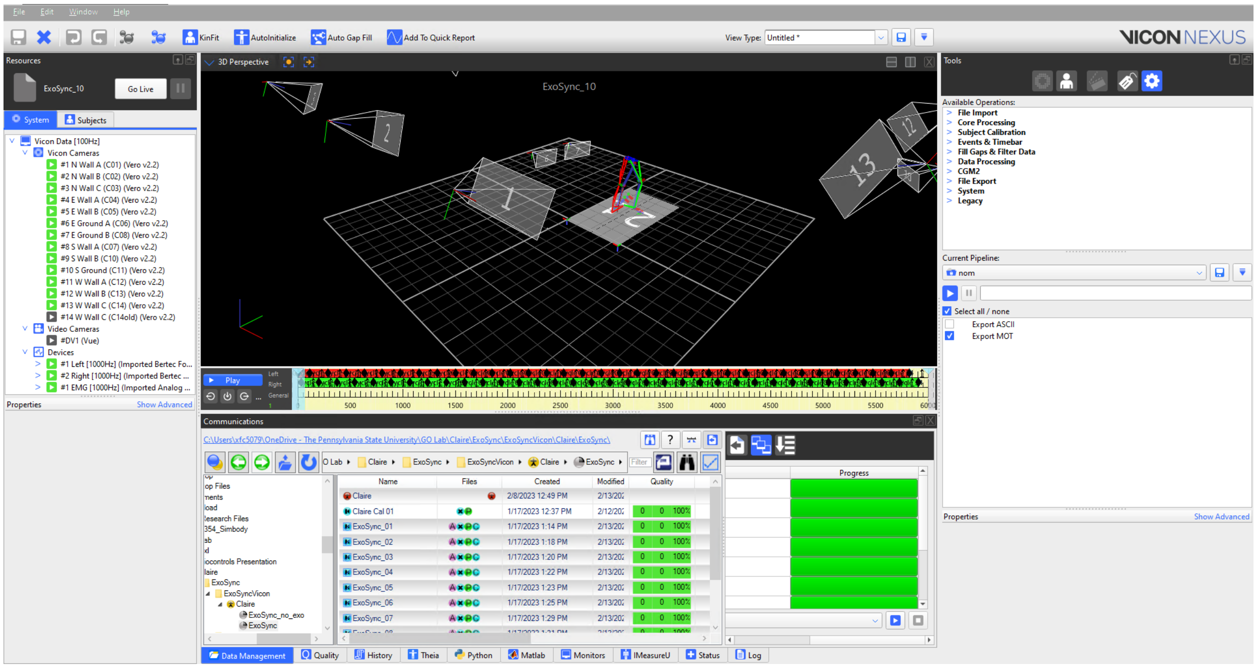
Preparation

Before you jump into the tutorial, make sure you have OpenSim downloaded at https://simtk.org/projects/opensim. Please also make sure you have went through the inverse kinematics (IK) tutorial, since an accurate set of IK data is very important to give appropriate ID results.

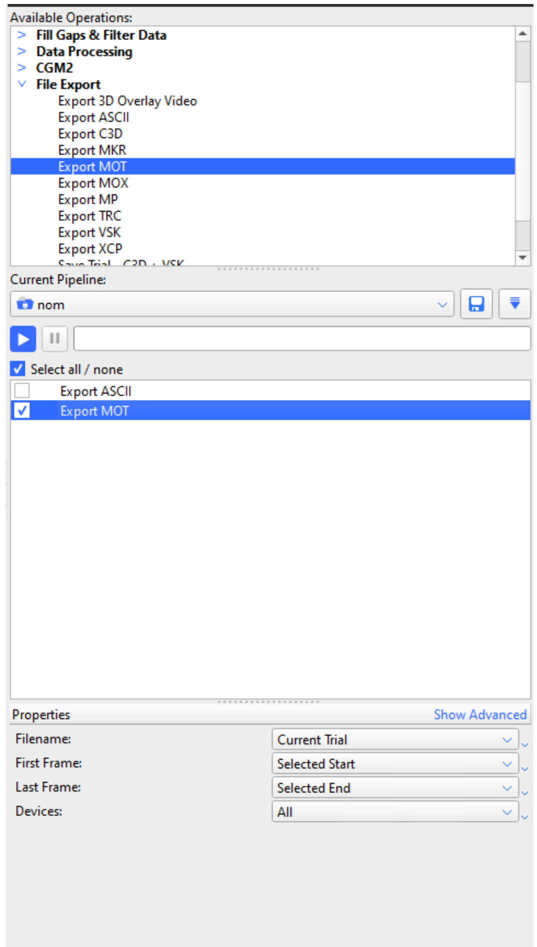
Please also make sure you have the subjects' pipelines went through in the Vicon Nexus. It is important that you have basic understanding how OpenSim's user interface works to operate the ID process.

1.Vicon Nexus File Exporting

The first step for setting up the ID processing is to get the ground reaction force (grf) data from Vicon Nexus. A typical Vicon trial interface should look like this:

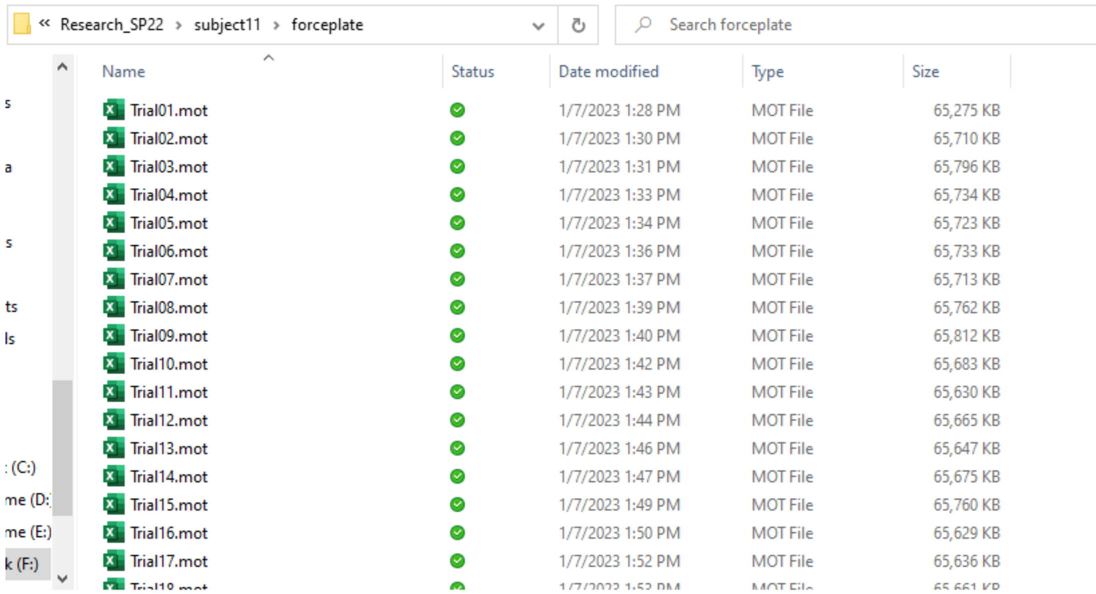


After you finished normal pipeline, which normally consists of marker labeling, gap filling, filtering trajectories, detecting events from forceplate, processing dynamic plug-in gait model, and calculation of gait cycle parameters, go to “File Export”, drag “Export MOT” into the current or a new pipeline, and export the grf data into .mot file.

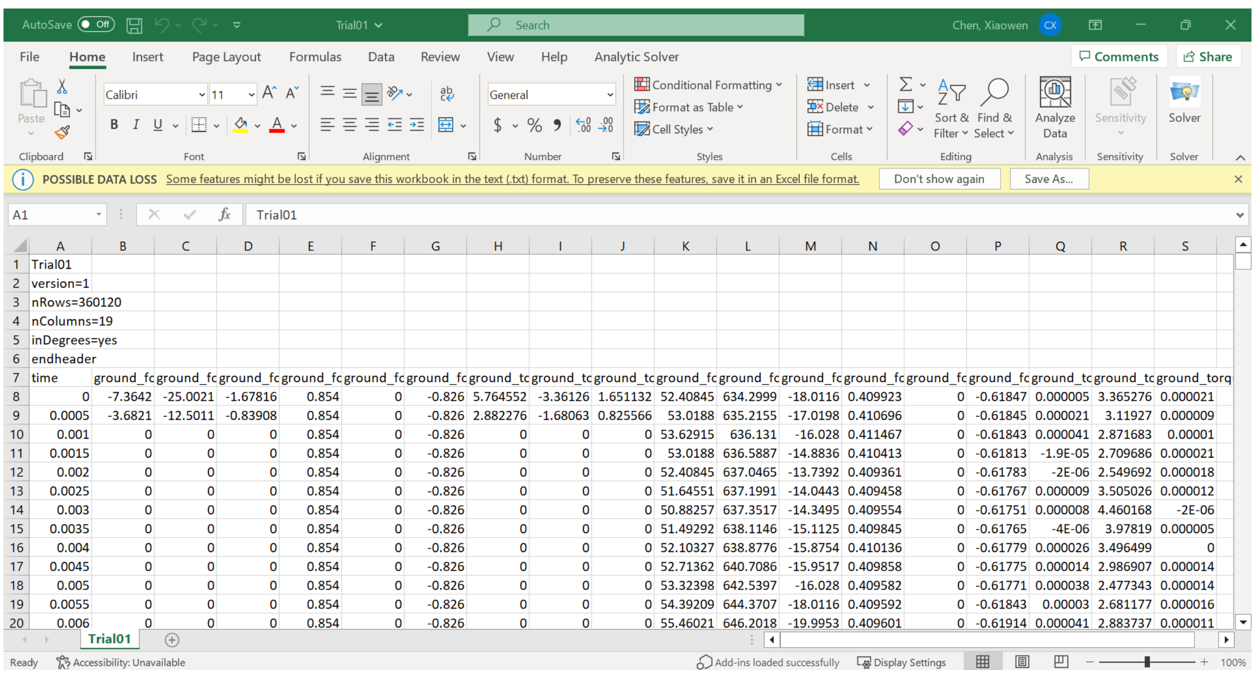


You can use the default setup for exporting or adjust the settings based on your needs.

After the .mot file are exported, please remember to move them to a new folder for ease of operations on OpenSim.



The exported .mot file should look like this:



	A	B	C	D	E	F	G	H	I	J
1	Trial01									
2	version=1									
3	nRows=360120									
4	nColumns=19									
5	inDegrees=yes									
6	endheader									
7	time	ground_fc	ground_fc	ground_fc	ground_fc	ground_fc	ground_fc	ground_tc	ground_tc	ground_tc
8	0	-7.3642	-25.0021	-1.67816	0.854	0	-0.826	5.764552	-3.36126	1.651132
9	0.0005	-3.6821	-12.5011	-0.83908	0.854	0	-0.826	2.882276	-1.68063	0.825566
10	0.001	0	0	0	0.854	0	-0.826	0	0	0
11	0.0015	0	0	0	0.854	0	-0.826	0	0	0
12	0.002	0	0	0	0.854	0	-0.826	0	0	0
13	0.0025	0	0	0	0.854	0	-0.826	0	0	0
14	0.003	0	0	0	0.854	0	-0.826	0	0	0
15	0.0035	0	0	0	0.854	0	-0.826	0	0	0
16	0.004	0	0	0	0.854	0	-0.826	0	0	0
17	0.0045	0	0	0	0.854	0	-0.826	0	0	0
18	0.005	0	0	0	0.854	0	-0.826	0	0	0
19	0.0055	0	0	0	0.854	0	-0.826	0	0	0
20	0.006	0	0	0	0.854	0	-0.826	0	0	0

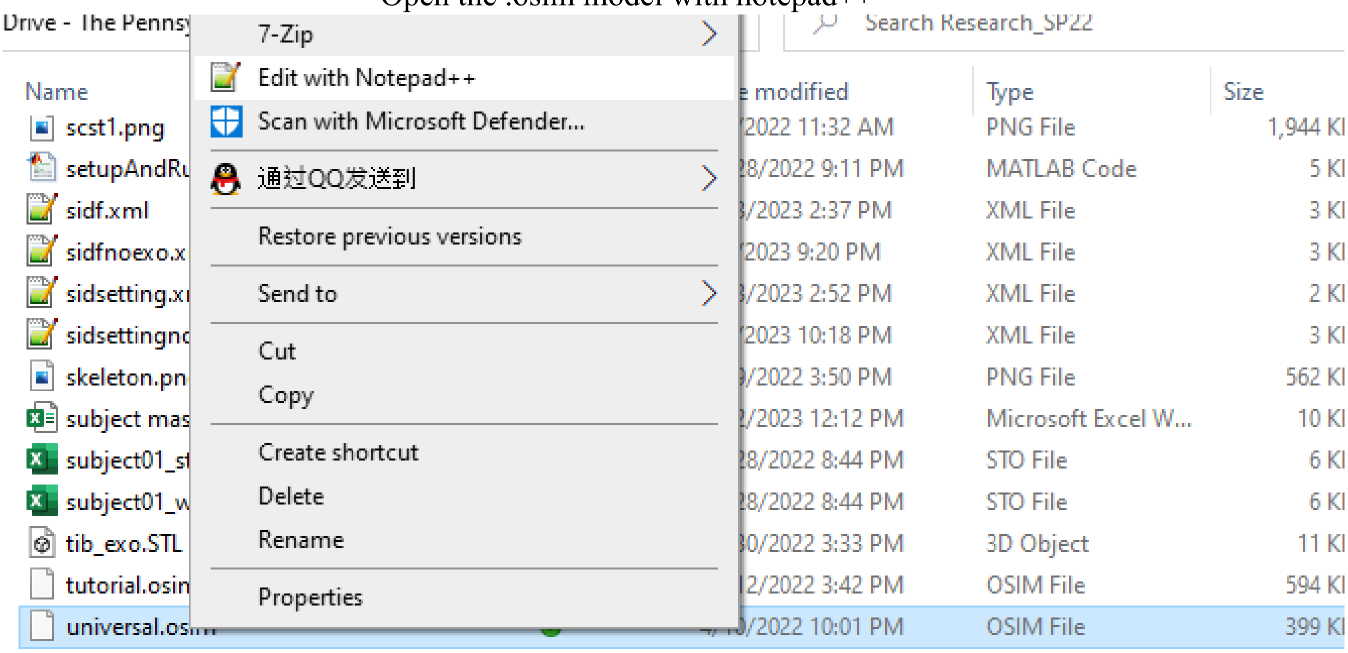
	K	L	M	N	O	P	Q	R	S
7	ground_fc	ground_fc	ground_fc	ground_fc	ground_fc	ground_fc	ground_tc	ground_tc	ground_torq
8	52.40845	634.2999	-18.0116	0.409923	0	-0.61847	0.000005	3.365276	0.000021
9	53.0188	635.2155	-17.0198	0.410696	0	-0.61845	0.000021	3.11927	0.000009
10	53.62915	636.131	-16.028	0.411467	0	-0.61843	0.000041	2.871683	0.00001
11	53.0188	636.5887	-14.8836	0.410413	0	-0.61813	-1.9E-05	2.709686	0.000021
12	52.40845	637.0465	-13.7392	0.409361	0	-0.61783	-2E-06	2.549692	0.000018
13	51.64551	637.1991	-14.0443	0.409458	0	-0.61767	0.000009	3.505026	0.000012
14	50.88257	637.3517	-14.3495	0.409554	0	-0.61751	0.000008	4.460168	-2E-06
15	51.49292	638.1146	-15.1125	0.409845	0	-0.61765	-4E-06	3.97819	0.000005
16	52.10327	638.8776	-15.8754	0.410136	0	-0.61779	0.000026	3.496499	0
17	52.71362	640.7086	-15.9517	0.409858	0	-0.61775	0.000014	2.986907	0.000014
18	53.32398	642.5397	-16.028	0.409582	0	-0.61771	0.000038	2.477343	0.000014
19	54.39209	644.3707	-18.0116	0.409592	0	-0.61843	0.00003	2.681177	0.000016
20	55.46021	646.2018	-19.9953	0.409601	0	-0.61914	0.000041	2.883737	0.000011

2. OpenSim Model Adjusting (Optional)

If the research has other objects attached to the subject, for example, an exoskeleton, you need to modify the .osim model file to make the model fit the real scenario.

To begin with, you should have notepad++ or other code editor installed on your local computer. To install notepad++, go to https://notepad-plus-plus.org/downloads/.

Open the .osim model with notepad++

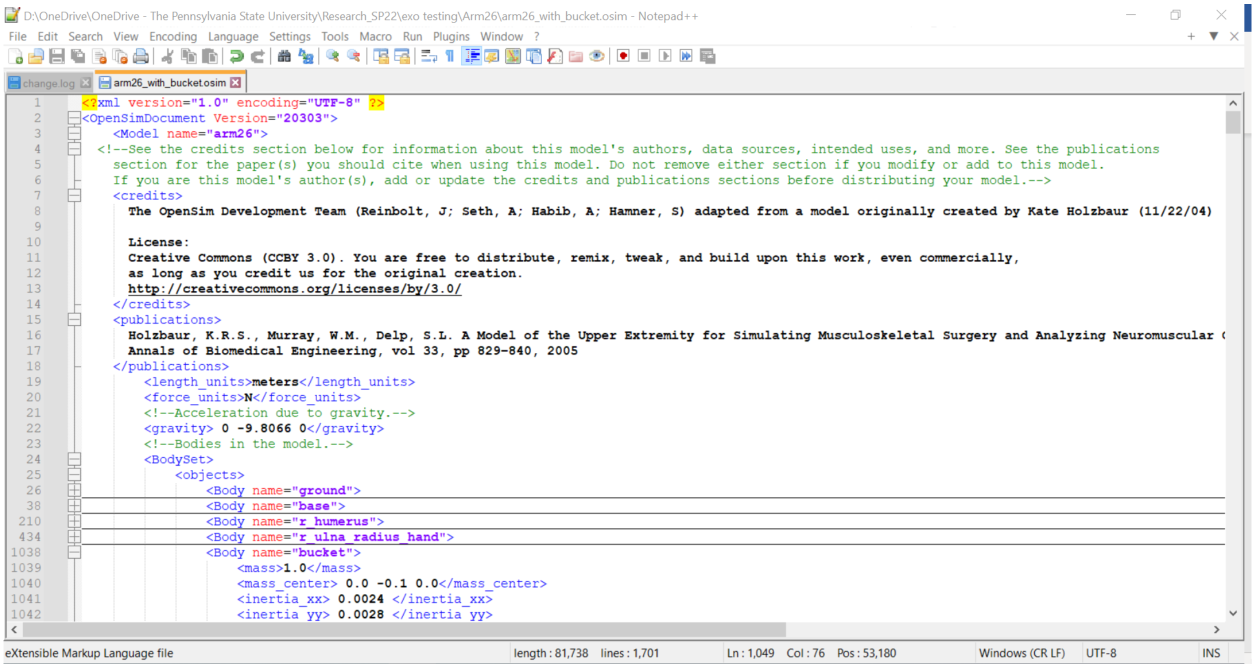


There is an official online example tutorial that teaches how to add a bucket dimension to an OpenSim .osim model. The link is as follows: https://simtk-confluence.stanford.edu:8443/display/OpenSim24/Example+-+Model+Editing.

Note1: the tutorial was followed, and the model was tested on OpenSim 4.3, but OpenSim 4.3 failed to have the model properly loaded. This suggests that the tutorial is not 100% accurate and contains misleading information.

Note2: only the model that has not been scaled can be modified. Do not try to add geometries to models that were already scaled down.

For correct codes of how to add a bucket to the model that can be properly loaded in OpenSim 4.3 and 4.4, please refer to Appendix A. Notice that the codes are only tested for OpenSim 4.3 and 4.4 and might not work in future versions.



The OpenSim model can read .stl and .vtp files as geometries. However, the .stl files are 1000 times larger than they're supposed to be when opened by OpenSim. Therefore, when you are loading .stl file into your model, make sure to revise this line:

<scale_factors> 1 1 1</scale_factors>

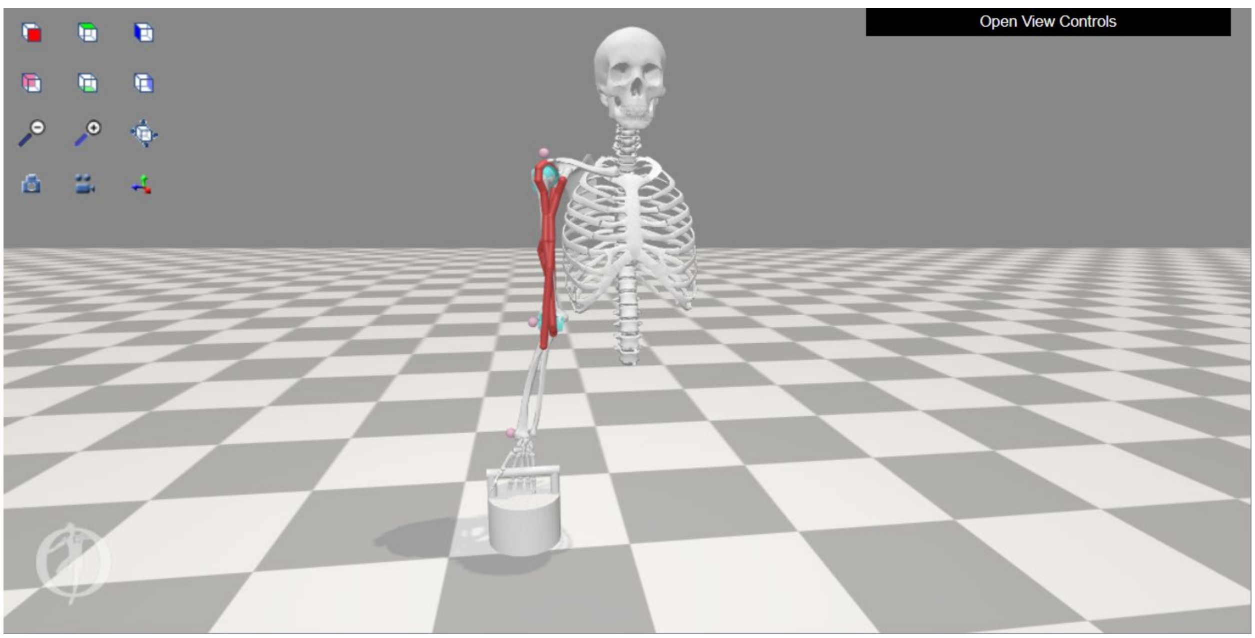
to

<scale_factors> 0.001 0.001 0.001</scale_factors>

Please make sure to measure the mass properties of the geometries you created in SolidWorks or other CAD tools with proper materials applied to them.

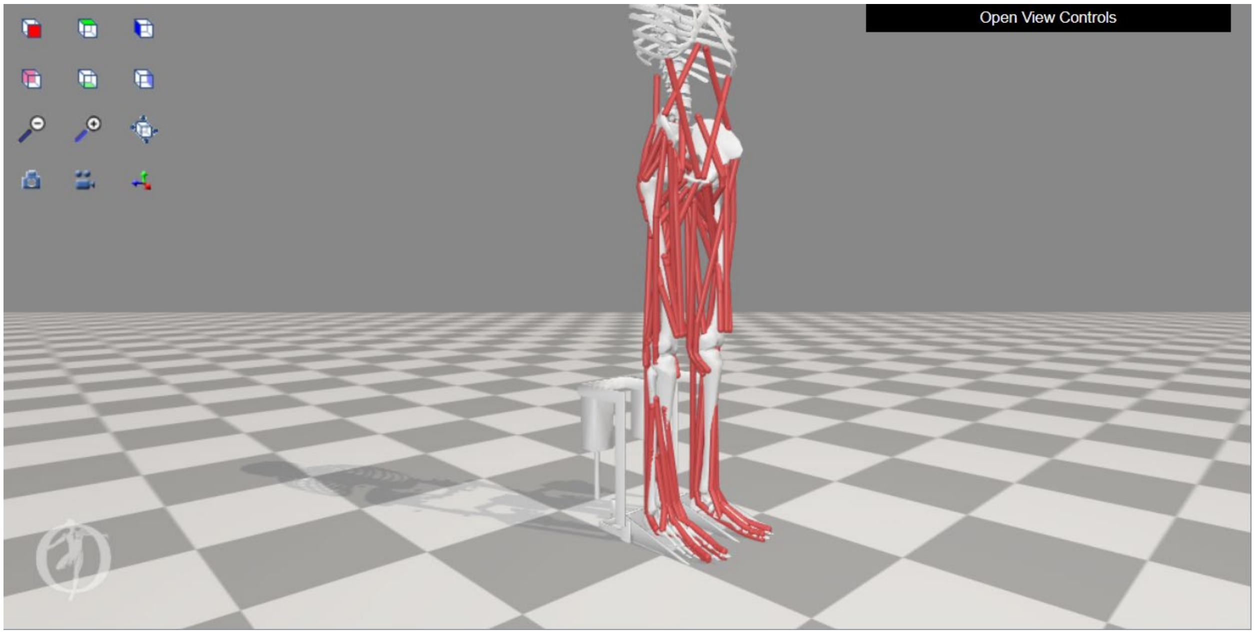
```xml
            Holzbaur, K.R.S., Murray, W.M., Delp, S.L. A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular (
            Annals of Biomedical Engineering, vol 33, pp 829-840, 2005
        </publications>
            <length_units>meters</length_units>
            <force_units>N</force_units>
            <!--Acceleration due to gravity.-->
            <gravity> 0 -9.8066 0</gravity>
            <!--Bodies in the model.-->
            <BodySet>
                <objects>
                    <Body name="ground">
                    <Body name="base">
                    <Body name="r_humerus">
                    <Body name="r_ulna_radius_hand">
                    <Body name="bucket">
                        <mass>1.0</mass>
                        <mass_center> 0.0 -0.1 0.0</mass_center>
                        <inertia_xx> 0.0024 </inertia_xx>
                        <inertia_yy> 0.0028 </inertia_yy>
                        <inertia_zz> 0.0024 </inertia_zz>
                        <inertia_xy> 0.0 </inertia_xy>
                        <inertia_xz> 0.0 </inertia_xz>
                        <inertia_yz> 0.0 </inertia_yz>
                        <Joint>
                            <PinJoint name="r_handle">
                                <parent_body> r_ulna_radius_hand </parent_body>
                                <location_in_parent> 0.031 -0.31 0.07 </location_in_parent>
                                <orientation_in_parent> 0.0 0.0 0.0 </orientation_in_parent>
                                <location> 0.0 0.0 0.0 </location>
                                <orientation> 0.0 0.0 0.0 </orientation>
                                <CoordinateSet name="">
                                    <objects>
                                        <Coordinate name="r_handle_rot">
                                            <motion type> rotational </motion type>
```

Once you have put the geometry codes into the OpenSim model, make sure you are able to see the newly added geometries.



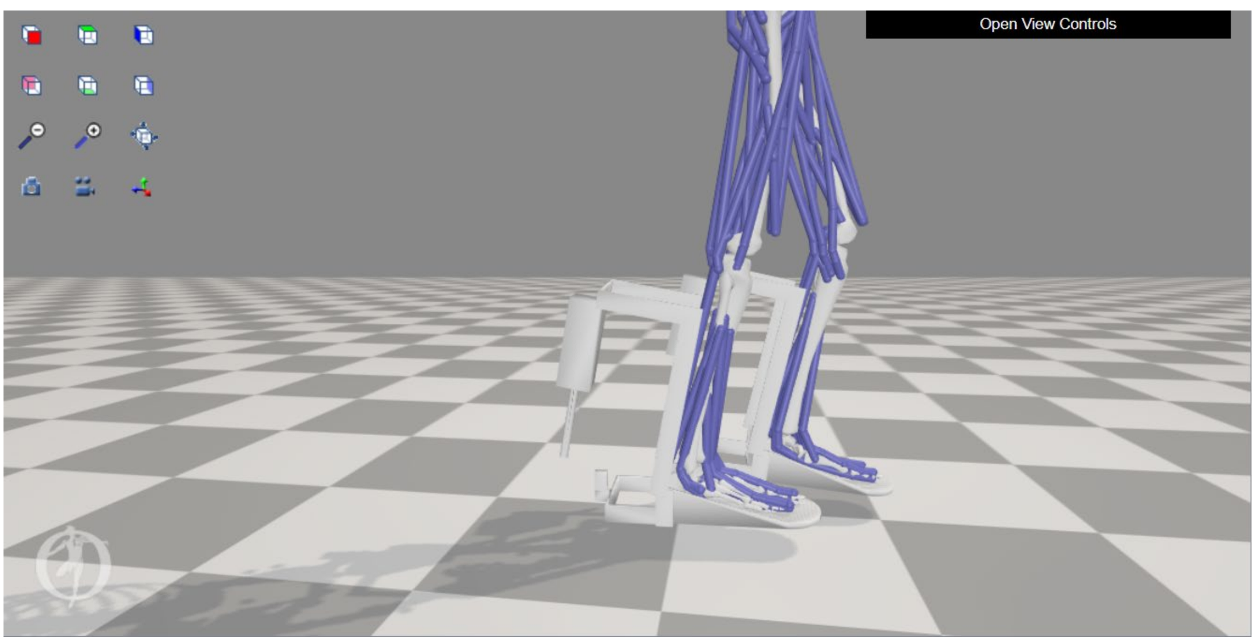
Please note that the geometry might not be in your wanted position, you can adjust the location, parent body, and location in parent accordingly until the geometry is in the correct location.

After you have learned how to adjust the model, you can use the tutorial as an example to adjust the exoskeleton model accordingly.



The exoskeleton model should have two parts: foot part and tibia part. Please attach them to the correct parent body accordingly.

After you have successfully put the models in there correct location, it is important for you to load a motion file or run an IK process to test if the model is attached correctly and running as expected.

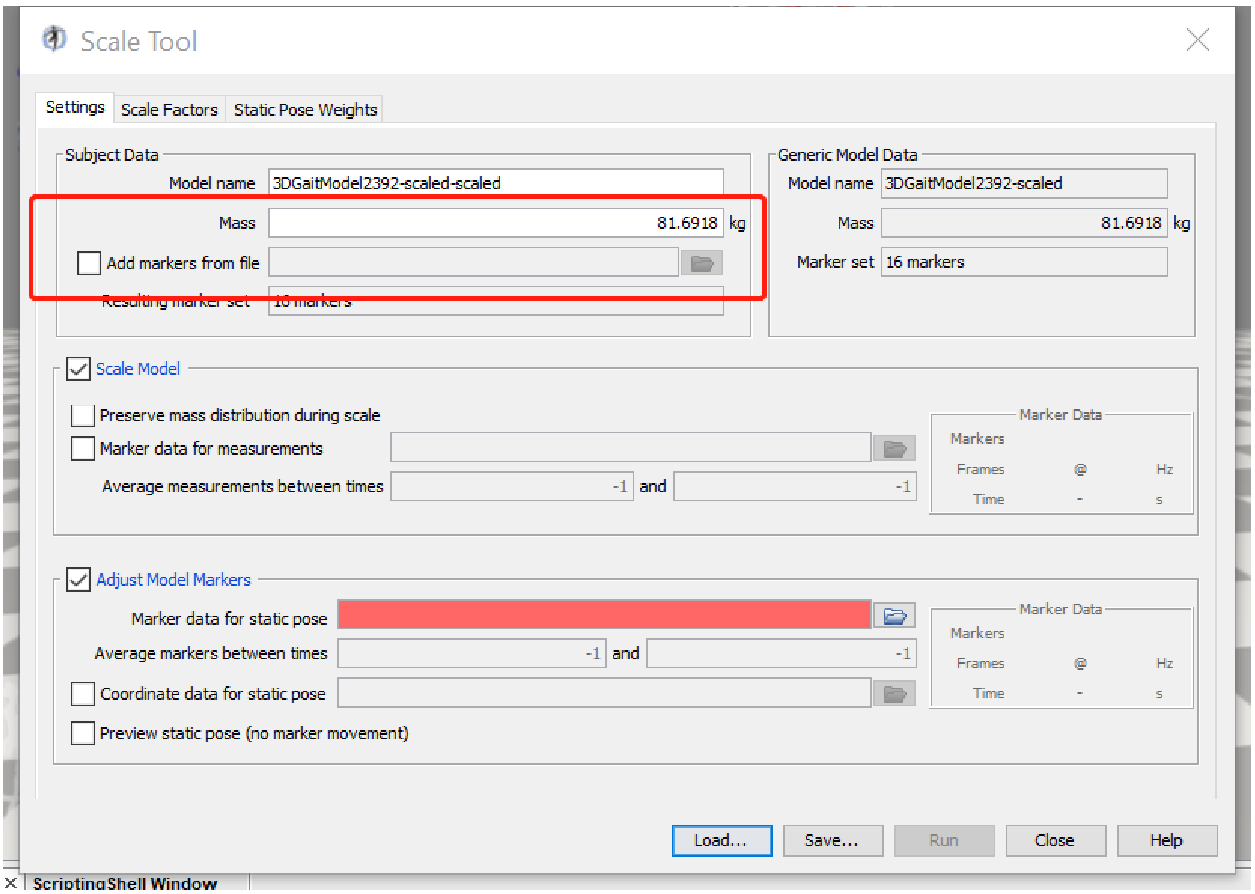


3. OpenSim ID Setup

Before you run the ID, make sure you have scaled down the musculoskeletal model to fit the subject's body geometries and weight.

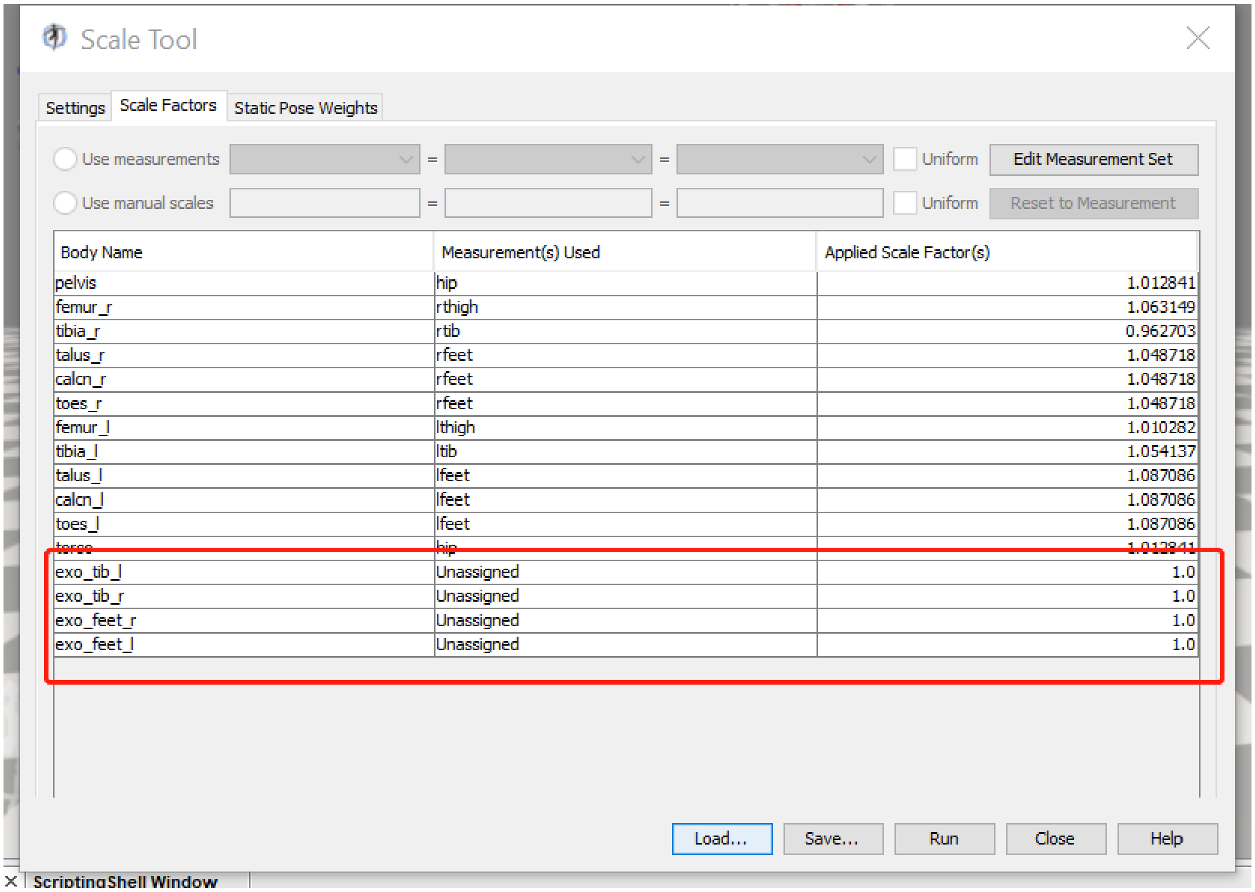
Please also remember to load the marker data when you are scaling it down. Please refer to section 3 in IK tutorial on how to save marker data.

The marker data can be loaded in OpenSim scale tool, where you can also change the body mass of the subject. Remember that if you are having subject wearing exoskeletons, you'll need to count the mass of the exoskeleton into the overall mass as well.



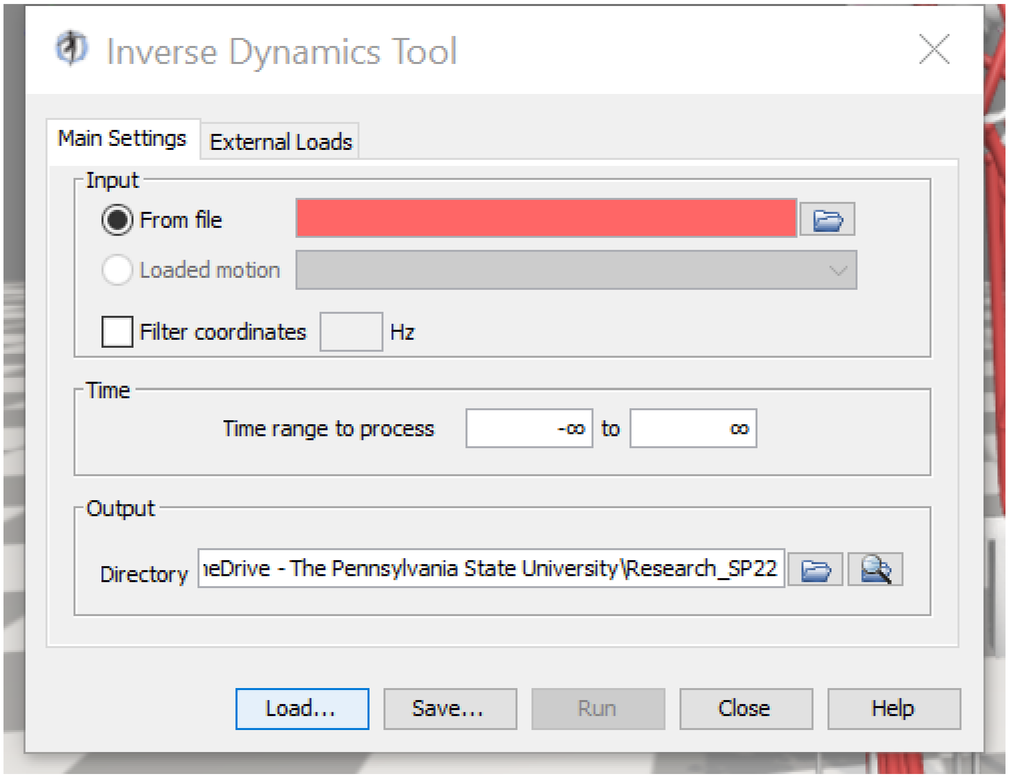
After you have finished marker adding and mass adjustments, you can scale the model following section 4 in IK tutorial.

Note that when you are having external geometries that will not change based on subject's body geometry, you need to assign a scaling factor of 1 to those geometries.



Please remember to save the scaling preset in case you want to go back and make adjustments to it.

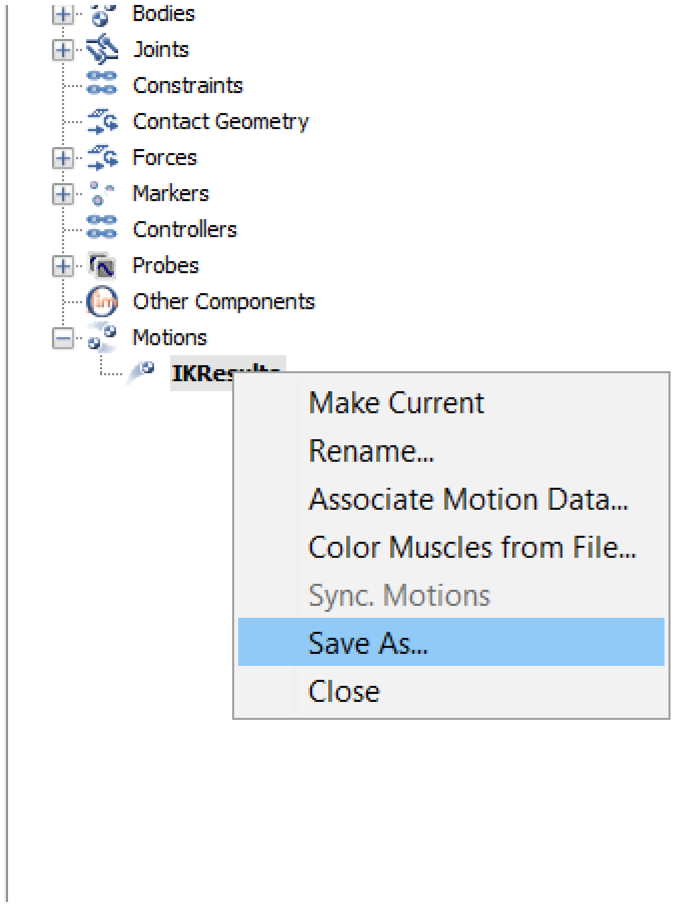
After you have finished scaling of the model, open up the “Inverse Dynamics” tool from OpenSim.



If you have already got an auto-generated file from MATLAB part, you can directly click "Load…" to load the preset file.

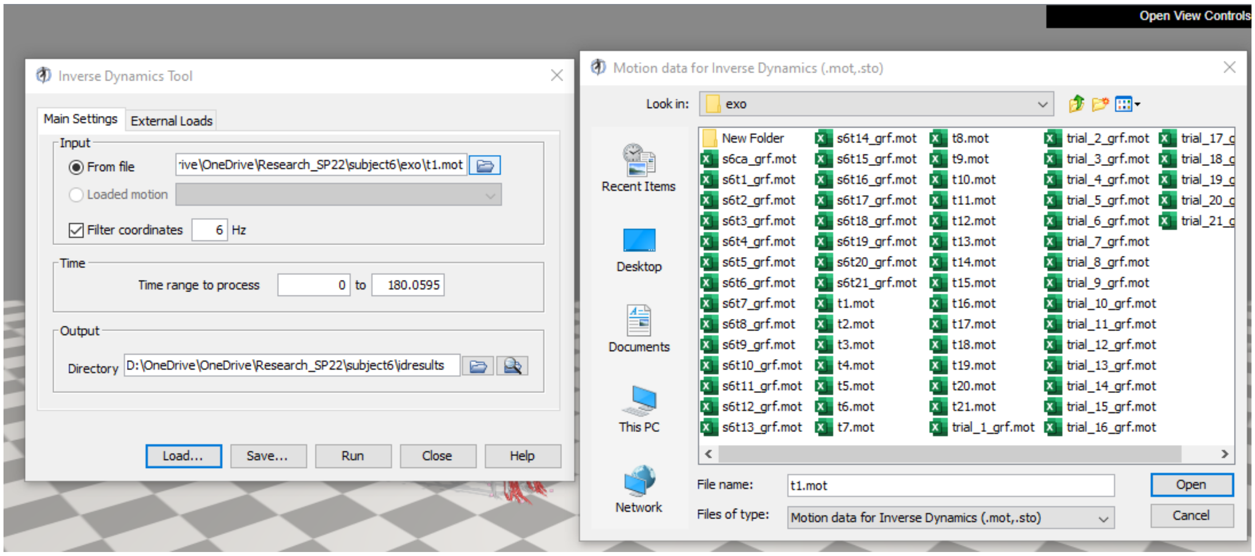
On the "Main Setting" page, you need to specify the input file, filtering frequency, ID starting and ending time, and output directory.

- The input file should be your prepared IK result file in .mot format. You can get this file by saving your IK results.

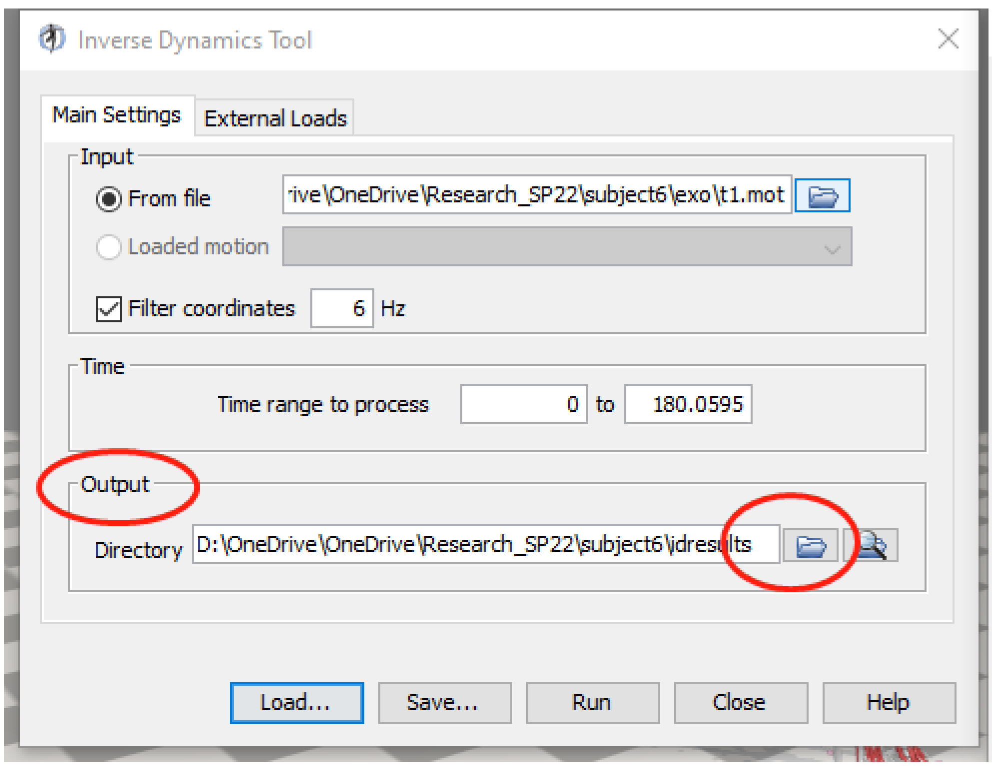


- Then, you need to prepare your grf file, which will also be in .mot format, as shown in section 1.
- The .mot grf file should be low-pass filtered without any phase change. You can use either a gaussian filter that is typically used in image processing techniques or filtfilt function in MATLAB. (To use the filtfilt function, please uncomment the filtfilt function part where you can find in the MATLAB codes, for detailed MATLAB codes, please see Appendix B. If you need to filter the data without running the entire MATLAB script, go to the note from section 4.2 for detailed instructions.)

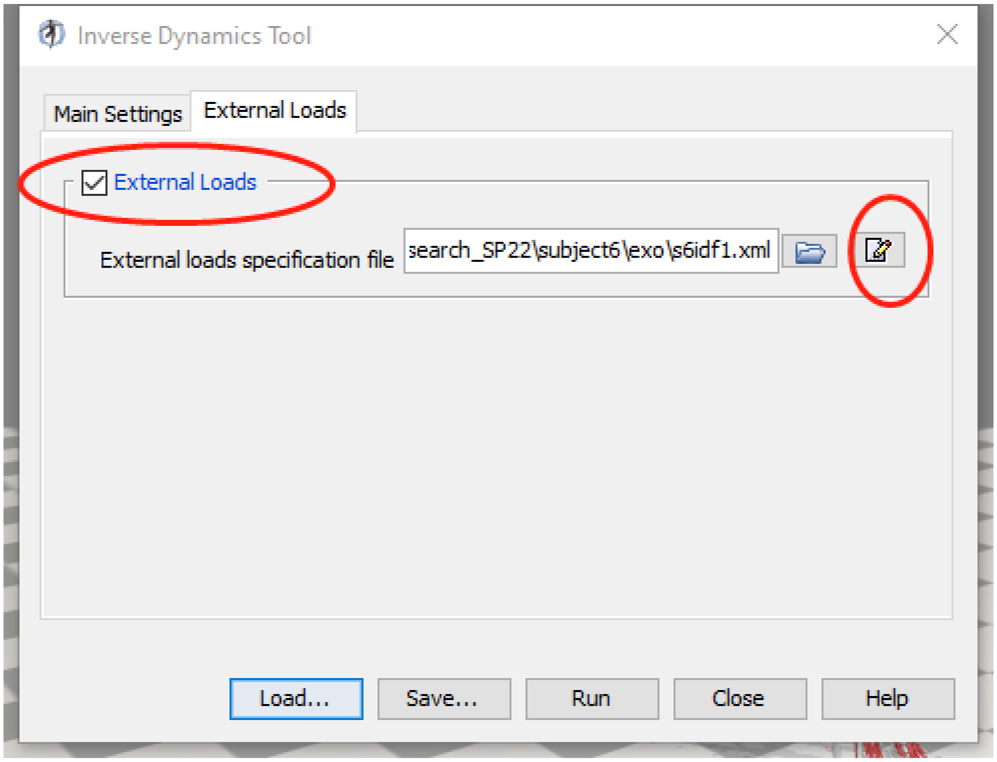
First, load the IK results file by selecting "From file" and select your IK results file from the file explorer.



Then, specify the filtering frequency and select the output directory by clicking on the folder icon under the “Directory” tab to select the folder you want to store your ID results.

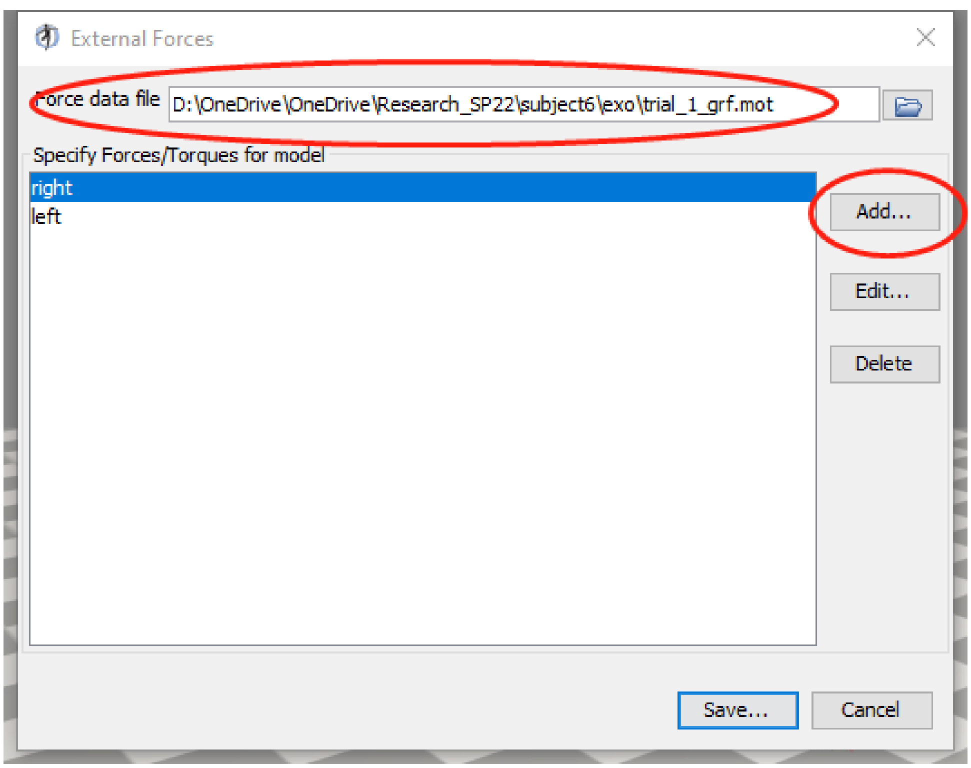


After you have set these up, click on the “External Loads” tab, check the “External Loads” checkbox, and click on the writing icon.

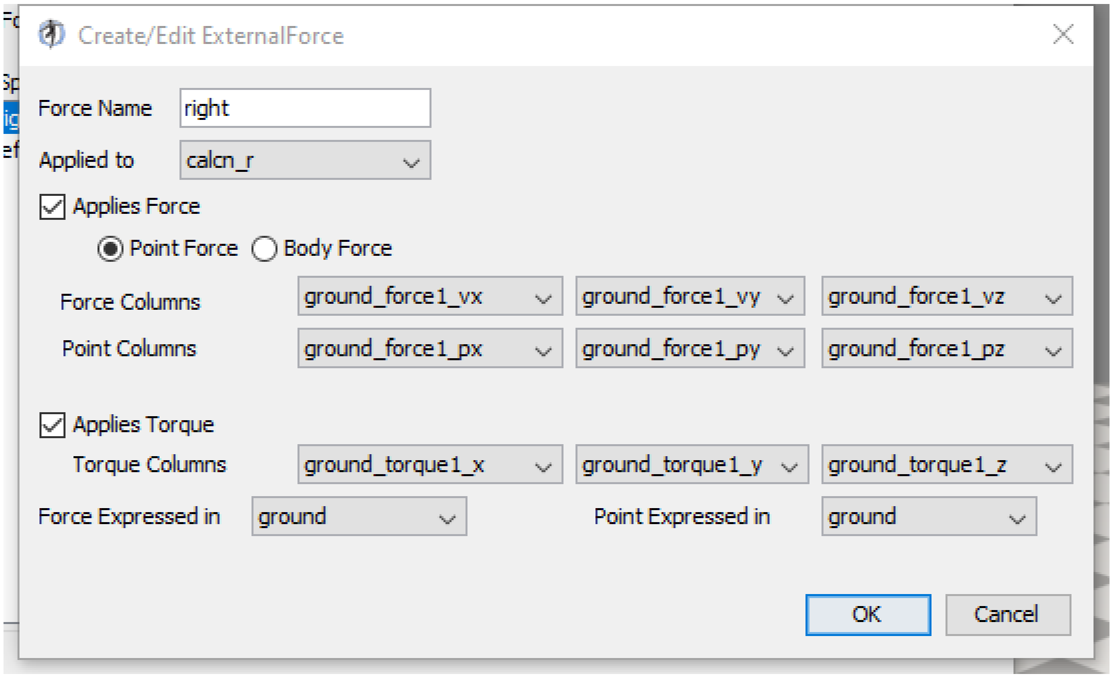


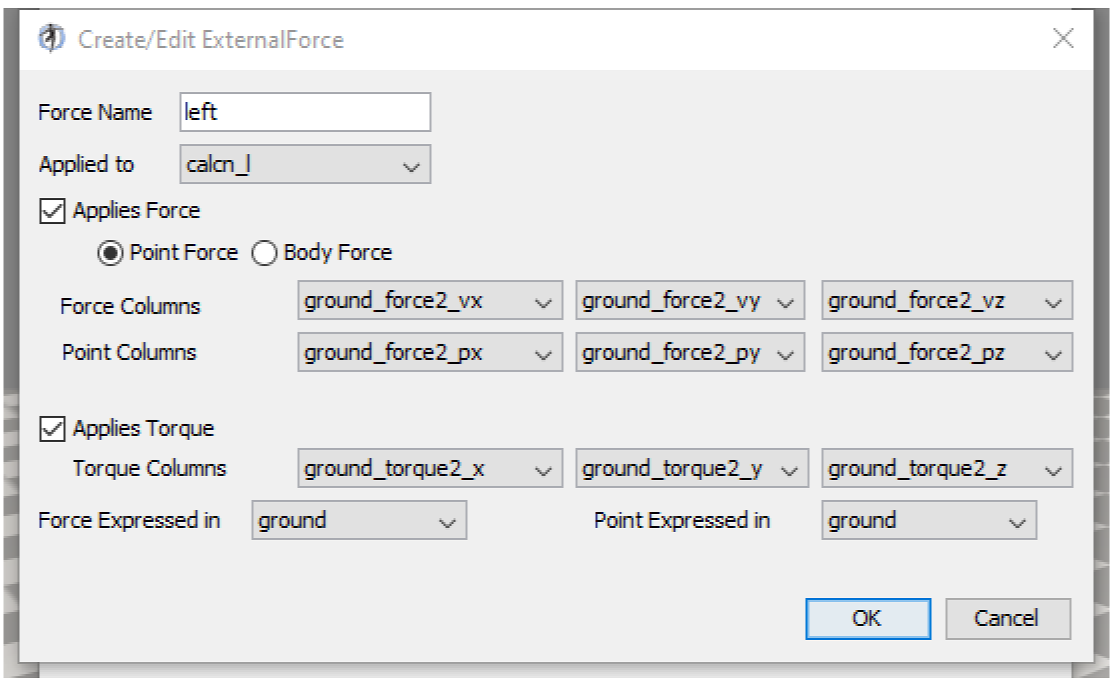
In the newly opened window, import the "Force data file" and select your low-pass filtered grf .mot file.

Click "Add" to specify forces and torques.



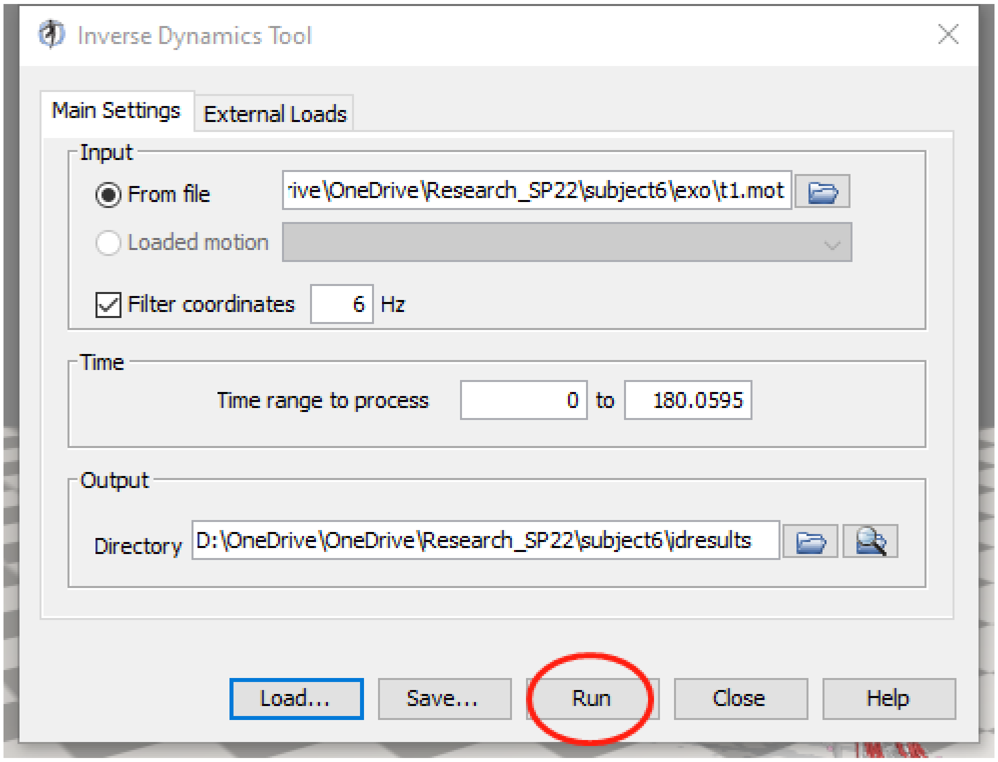
Set the left and right applied force as shown in the picture:





- NOTE!!!: the left and right external forces might be flipped, please compare your grf data with your IK results and Vicon behaviors and check other resources to make sure that you have the corresponding columns input.

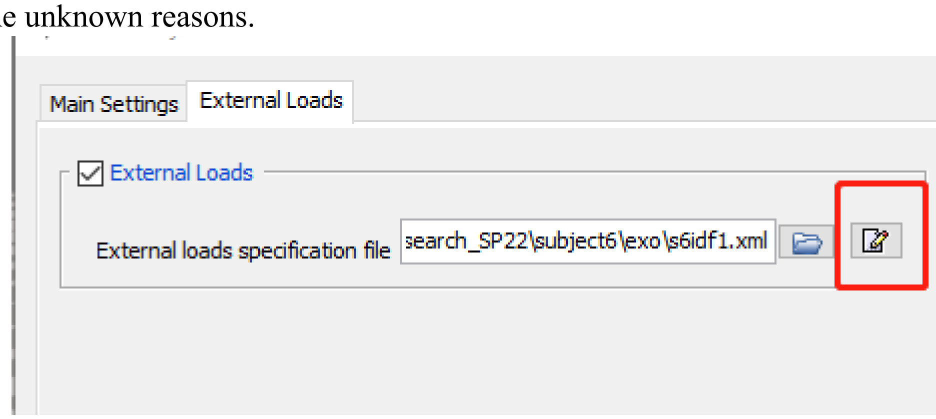
After you have made sure the inputs are all correct, hit “OK” for the opened windows, and on the ID tools, hit “Run”.



The OpenSim "Messages" window should show that the ID is running. If you encountered any errors, please go through section 1 to 3 again to ensure you went through everything as directed.

Here are some common causes for OpenSim popping up errors:

1. The IK result and grf file are having different sampling frequencies or different starting and ending points, namely, their data sizes are not the same.
2. Data files are corrupted.
3. The IK result is not for the .osim model.
4. There are some well-known issues for creating/edit external force tools in OpenSim, please make sure that the forces are correctly loaded by checking it again. To check if the forces are correctly loaded, open the external loads again and verify that both left and right forces are correctly loaded. Notice that sometimes the loaded forces can disappear for some unknown reasons.



4. Batch Processing with MATLAB Script

4.1. Working Environments Setup

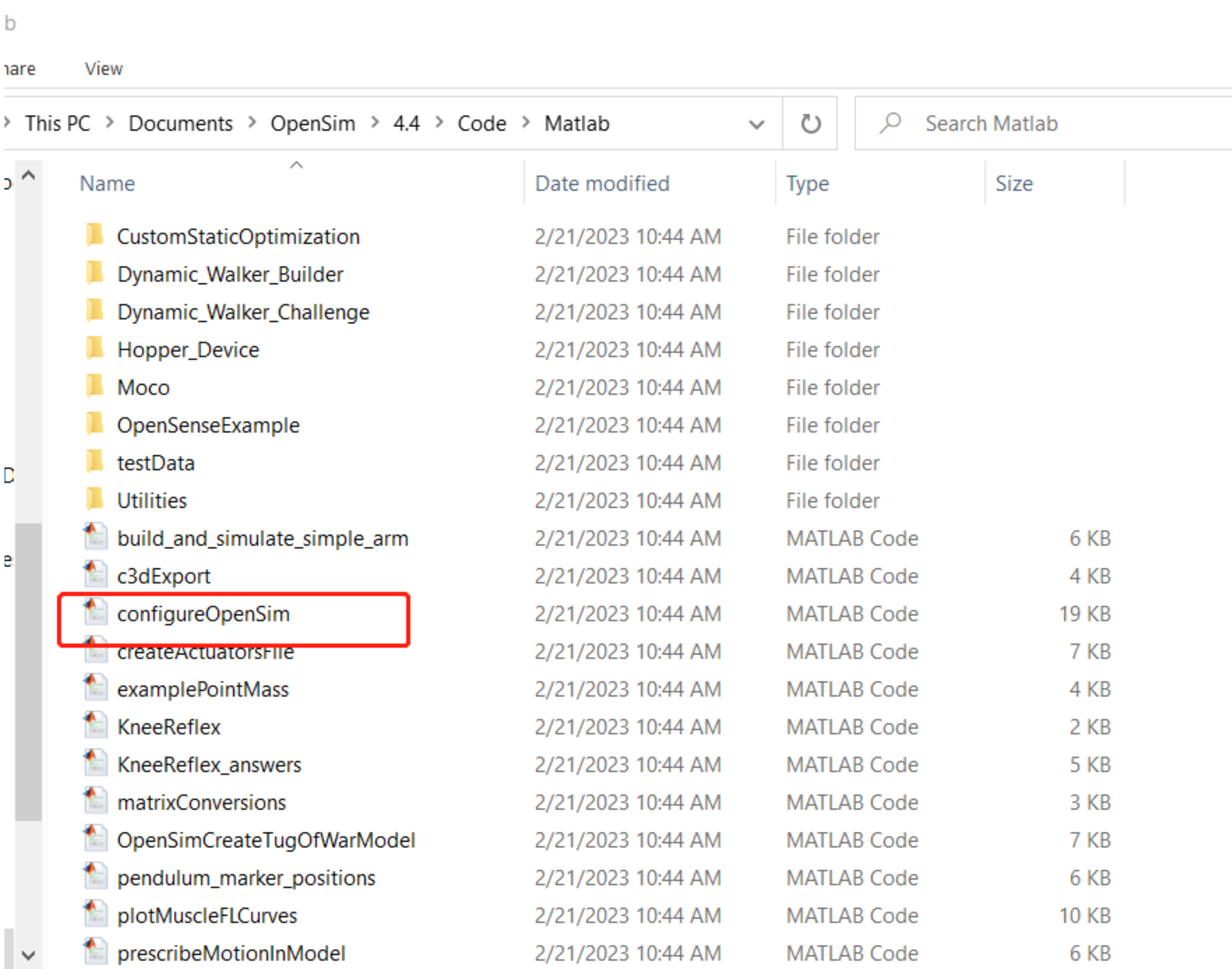
To view the entire script for the batch processing, please check Appendix B. Here, we'll talk about how to setup some of the most important prerequisites for the MATLAB script that we're using in this tutorial.

Before you run the codes, please make sure you have setup the required environments for running the codes. The requirements for running the code include OpenSim working environment, Jonathan's MoCapTools, and James Conder's Gaussian Filter script.

To setup the OpenSim's working environment, you can follow the instructions on the OpenSim's official website: https://simtk-confluence.stanford.edu:8443/display/OpenSim/Scripting+with+Matlab

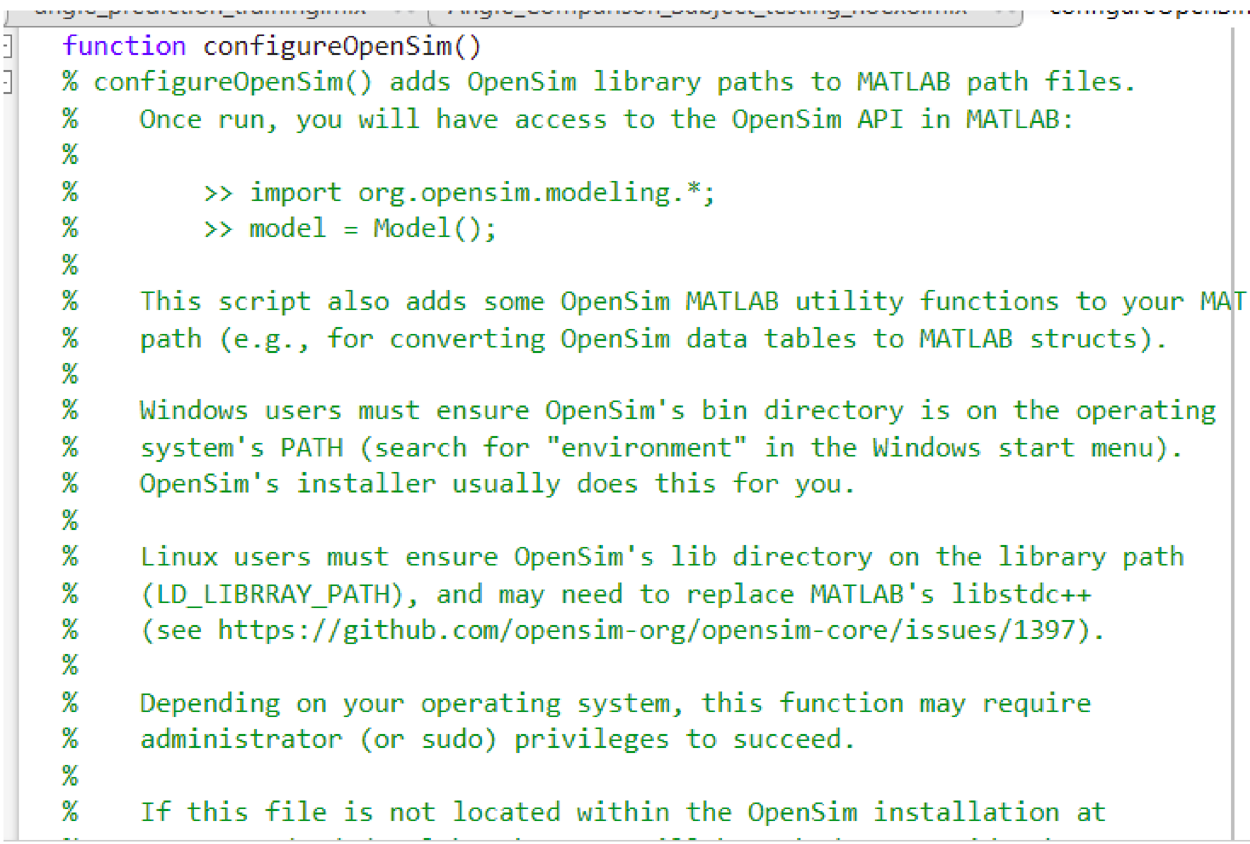
To get started, locate the OpenSim's official MATLAB codes from C:\Users\<your user name>\Documents\OpenSim\4.4\Code\Matlab

Under the folder, find the code named "configureOpenSim"



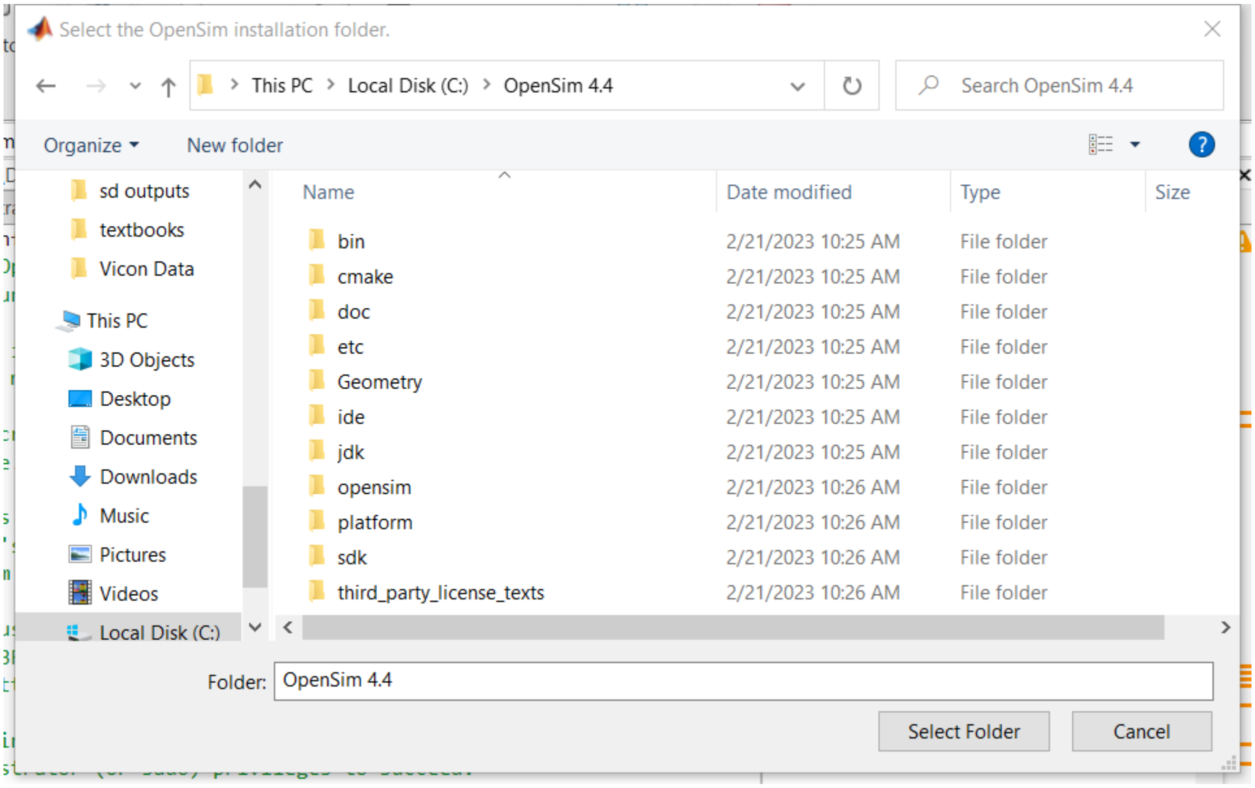
(If you do not have the matlab code, make sure you are under the correct windows user directory as it might appear in someone else's document folder. Also make sure you are installing OpenSim correctly. If you still failed to locate the codes, please reinstall the OpenSim)

Open the code, it should look like this:



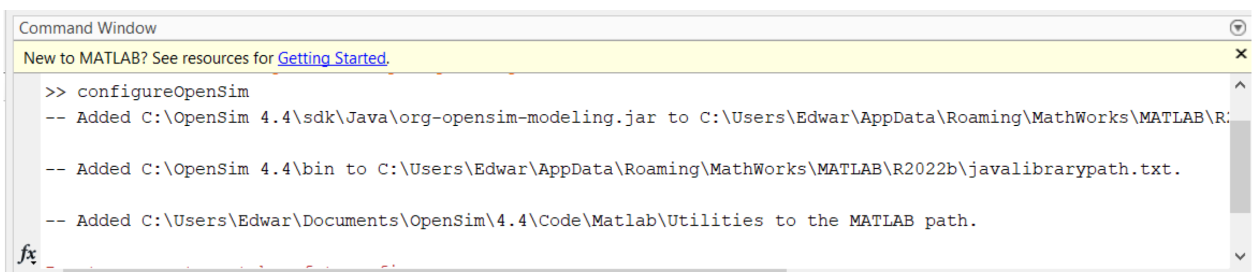
```
function configureOpenSim()
% configureOpenSim() adds OpenSim library paths to MATLAB path files.
%    Once run, you will have access to the OpenSim API in MATLAB:
%
%        >> import org.opensim.modeling.*;
%        >> model = Model();
%
%    This script also adds some OpenSim MATLAB utility functions to your MAT
%    path (e.g., for converting OpenSim data tables to MATLAB structs).
%
%    Windows users must ensure OpenSim's bin directory is on the operating
%    system's PATH (search for "environment" in the Windows start menu).
%    OpenSim's installer usually does this for you.
%
%    Linux users must ensure OpenSim's lib directory on the library path
%    (LD_LIBRRAY_PATH), and may need to replace MATLAB's libstdc++
%    (see https://github.com/opensim-org/opensim-core/issues/1397).
%
%    Depending on your operating system, this function may require
%    administrator (or sudo) privileges to succeed.
%
%    If this file is not located within the OpenSim installation at
```

Hit "Run", and the code will let you choose your OpenSim's installation directory, make sure you select the correct folder.



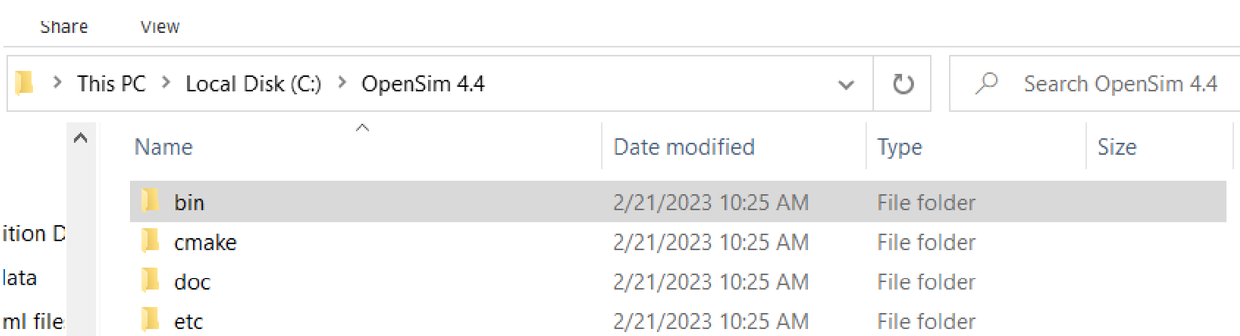
By default, the OpenSim should be installed under C:\OpenSim X.X

After you selected the folder, the OpenSim will be configured for your computer, and you should see the confirmation of paths being added in your MATLAB command window.

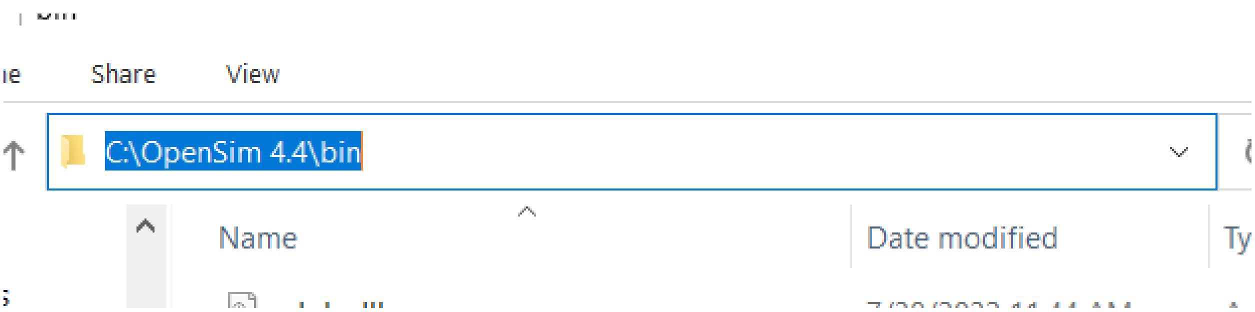


After you configured OpenSim, you should add OpenSim to your environment variables.

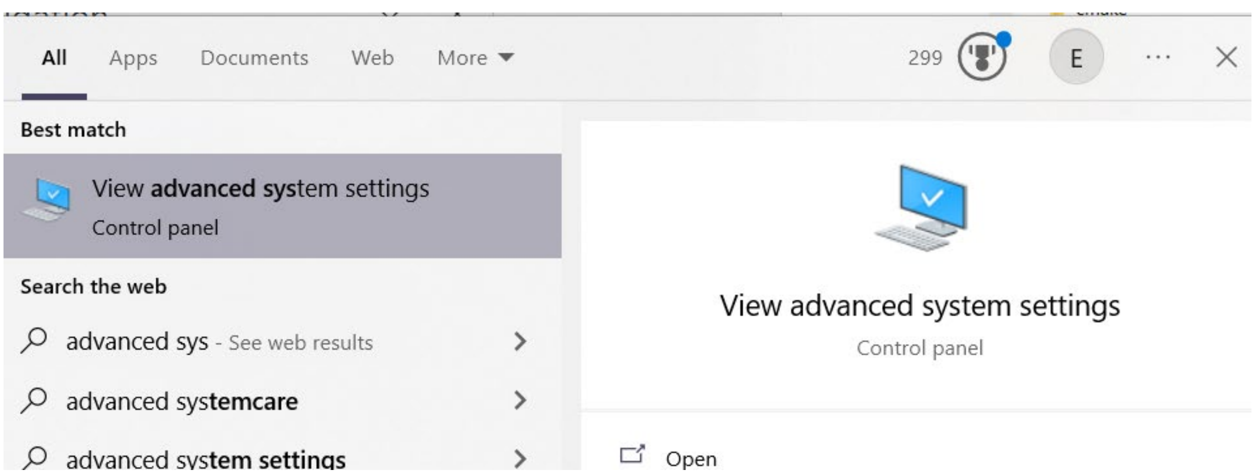
First, locate the "bin" folder under your OpenSim installation directory.



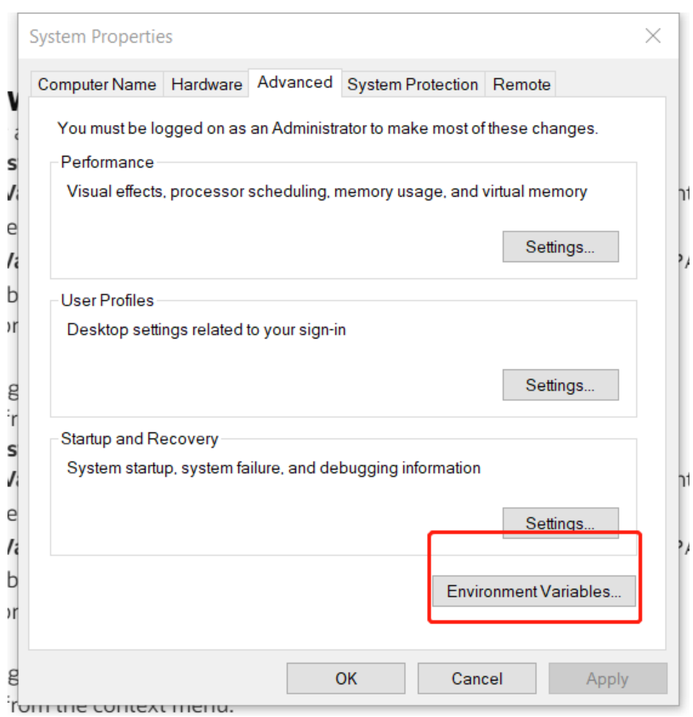
Copy the path to this folder



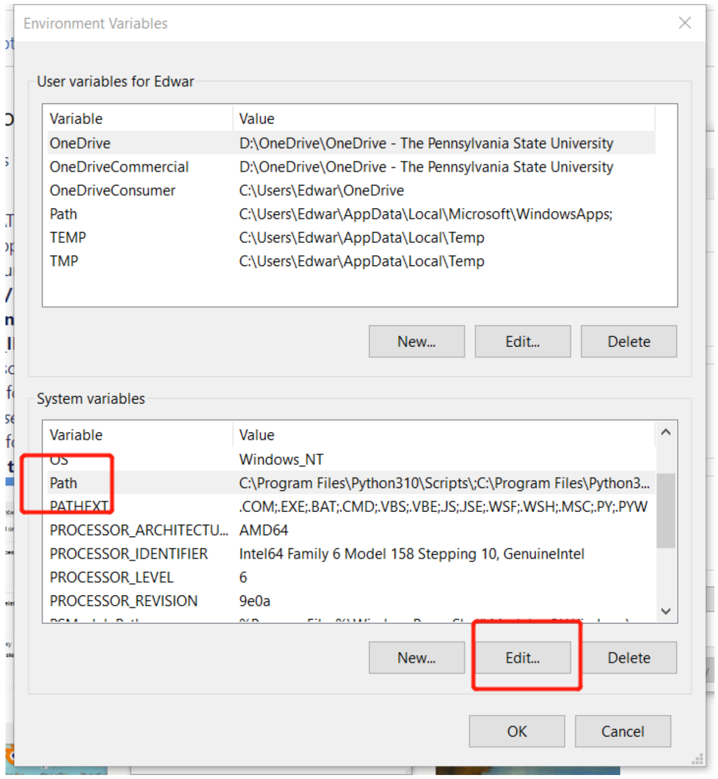
Then, go to your search box, and type “advanced system settings” and locate it in the control panel.



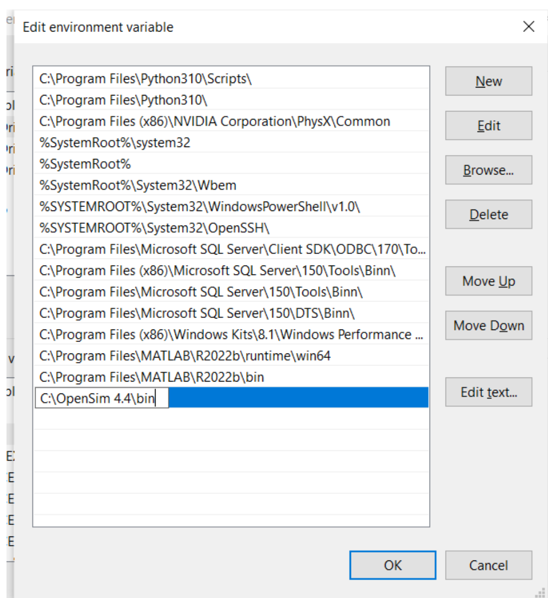
After you opened it, click “environmental variables”



Under “system variables”, click “Path”, and click “Edit”



Then, in the new window, click “New”, and paste your copied path in there, and hit “OK”



Hit “OK” for the following windows, and you’re done with setting up the OpenSim’s working environment.

To setup the two toolbox we're going to use for the MATLAB script, go to the links:

https://github.com/JonathanCamargo/MoCapTools.git

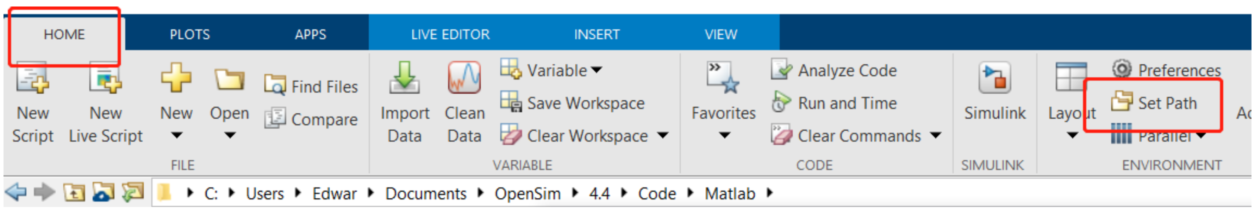
https://www.mathworks.com/matlabcentral/fileexchange/43182-gaussfilt-t-z-sigma?s_tid=prof_contriblnk

download both the scripts to your MATLAB document folder, you can find the folder in

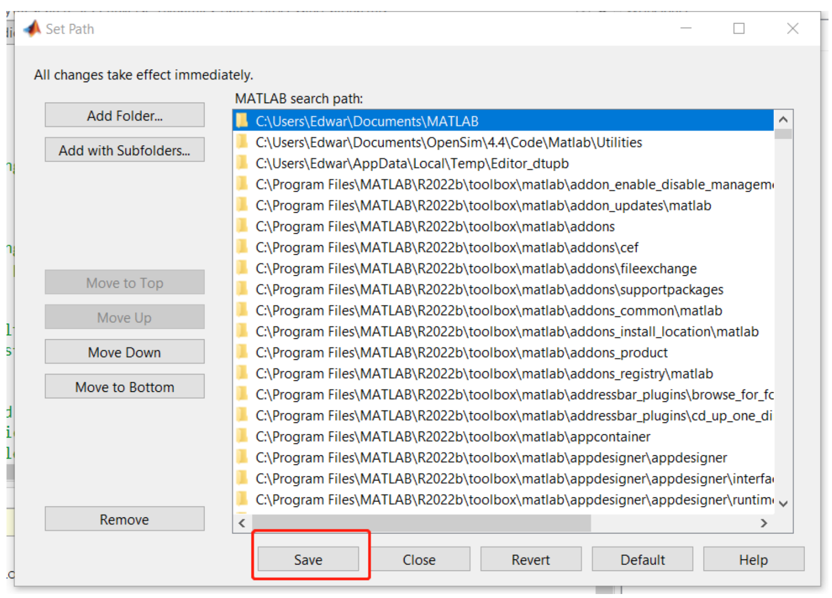
C:\Users\<Your User Name>\Documents\MATLAB

After you've put the folders in there, please also make sure to add the scripts to path using the following steps:

Go to Home→Set Path



Click "Add Folder", and select the MATLAB folder, then click "Save"



Once the paths are added to the folder, the environmental setting is done.

4.2. Operating With the MATLAB Script

The script, which is attached to the Appendix B, can be separated into 5 sections.

The first section is line 1-54, where the batch processing will setup the folders and files for operations. During this section, there will be multiple prompt windows asking for your folders and file data. Before you run this section, please make sure that you have went through the entire section 1 and exported your grf and IK results data.

The second section ranges from 55-123, where the codes will process the filtering of the grf data.

- NOTE: If you are looking for filtering the data without running the entire MATLAB code, copy and paste part of the second code section, and use some file management methods in the first code section to help you pass through the filtering process. This tutorial will not go through the entire process but the code comments are already detailed enough to help you setup your own script.

The third section is from line 124 to 135, it will copy and modify the preset ID grf .xml setup file to your IK results folder and use them for ID batch processing.

The fourth section, located between line 136 and 153, will copy and modify the preset ID processing .xml setup file to your IK results for ID batch processing. You can also use the copied preset file for ID processing in OpenSim UI by clicking “Load” in the ID tool.

- NOTE: The two .xml file is attached in Appendix C, we will not go over them in detailed here. You can use OpenSim ID toolbox to generate those custom .xml files yourself, but they might not work with this MATLAB script.

The fifth section will then setup the OpenSim environment and ask you for the .osim model that will be used for ID processing. If you have not setup a proper model, please go to section 3 and 4 in IK processing tutorial and section 2 in this tutorial to see how to setup an OpenSim model for ID processing.

If the fifth section shows up error, it can be caused by wrong data file sizes, wrong .osim model, or loss of OpenSim environment. Though it is very unlikely for the error to be shown as the previous codes will prompt many possible mistakes once if found it automatically, it is still possible that you encounter such circumstances. Please make sure you have carefully followed the tutorial, and if you still have any problems, please check if the data is corrupted or not as expected by the code.

- NOTE: The script attached in Appendix B is customized for plug-in gait research, where each set of data has its custom column. If you are doing other researches, it is possible that your grf data is different than the data that this script is intended to operate upon. Please revise the second section of the script until the data are assigned to their corresponding variables. You can use the ID external loads setup methods in section 3 to help you understand how to assign the columns to the variables.

Once the MATLAB script finishes and no error pops up, you can go to your designated results folder and check if the data generated is as expected. Please compare a set of sample result with other result sources (e.g., Vicon joint moments results) to make sure that the result you get is accurate and corresponding IK and grf data files are assigned to the ID processing.

Appendix A: Codes of Adding Bucket to Model

These codes are built based on OpenSim online tutorial Example-Model Editing: https://simtk-confluence.stanford.edu:8443/display/OpenSim24/Example+-+Model+Editing.

This code only works if you have followed the online tutorial and created a new Arm26.osim file and found the correct location to place the codes. Please be aware that these codes are space sensitive.

Code Body (space sensitive):

```
<Body name="bucket">
        <mass>1.0</mass>
            <mass_center> 0.0 -0.1 0.0</mass_center>
            <inertia_xx> 0.0024 </inertia_xx>
            <inertia_yy> 0.0028 </inertia_yy>
            <inertia_zz> 0.0024 </inertia_zz>
            <inertia_xy> 0.0 </inertia_xy>
            <inertia_xz> 0.0 </inertia_xz>
            <inertia_yz> 0.0 </inertia_yz>
            <Joint>
              <PinJoint name="r_handle">
                    <parent_body> r_ulna_radius_hand </parent_body>
                        <location_in_parent> 0.031 -0.31 0.07 </location_in_parent>
                        <orientation_in_parent> 0.0 0.0 0.0 </orientation_in_parent>
                        <location> 0.0 0.0 0.0 </location>
                        <orientation> 0.0 0.0 0.0 </orientation>
                        <CoordinateSet name="">
                          <objects>
                                <Coordinate name="r_handle_rot">
```

<motion_type> rotational
</motion_type>
<default_value> 0.0
</default_value>
<default_speed_value> 0.0 </default_speed_value>
<initial_value> 0.0
</initial_value>
<range>-3.14159265
3.14159265 </range>
<clamped> false
</clamped>
<locked> false
</locked>
<prescribed_function/>
</Coordinate>
</objects>
<groups/>
</CoordinateSet>
</PinJoint>
<reverse> false </reverse>
</Joint>
<VisibleObject name="">
<GeometrySet>
<objects>
<DisplayGeometry>
<geometry_file>bucket.vtp</geometry_file>
<color> 1 1 1 </color>
<texture_file />

<transform> -0 0 -0 0 0
0</transform>
<scale_factors> 1 1
1</scale_factors>
<display_preference>4</display_preference>
<opacity>1</opacity>
</DisplayGeometry>
</objects>
<groups />
</GeometrySet>
</VisibleObject>
</Body>

Appendix B: MATLAB Codes for ID Batch Processing

```
 clear,clc
 % This batch processing code targets to use OpenSim's built-in inverse
 % dynamics function to get joint moment data for subject trials.

 % Auther: Xiaowen Chen
 % Plug-in Gait Optimization Lab
 % Mechanical Engineering Department
 % Pennsylvania State Universeity

 % Part of the code is built from OpenSim official matlab scripts:
 % https://simtk-
confluence.stanford.edu:8443/display/OpenSim/Scripting+with+Matlab

 % ################## Before you run the code ##############
 % Setup OpenSim working environment:
 % https://simtk-
confluence.stanford.edu:8443/display/OpenSim/Scripting+with+Matlab
 % Install Jonathan's MoCapTools and add the folder and subfolders to path
 % https://github.com/JonathanCamargo/MoCapTools.git
 % Install James Conder's Gaussian Filter Script and add to path
 % https://www.mathworks.com/matlabcentral/fileexchange/43182-gaussfilt-t-z-
sigma?s_tid=prof_contriblnk
 % (if you don't want to use gaussian filter, please comment the gaussfilt()
 % function and uncomment the filtfilt() function below it.

 % Please make sure that the two .xml setup files named 'sidf.xml' and
 % 'sidsetting.xml' are downloaded. Please finish the inverse kinematics of
 % the trials if you have not done so and plot the hip, knee, and ankle
 % joint angles and exported them. Also make sure that the .mot file that
 % contains the ground reaction force is available before running this code.
 % ## Please make sure that there are no more .mot files other than the grf
 % data.
 % ## Please also make sure that you name the Inverse Kinematics Trials data
 % in format of t*ao, for example, t1ao, t2ao, ..., t10ao.
 % ## Please make sure each .mot file corresponds to an inverse kinematics data
file
 % ##############################################################


 subjectdir = uigetdir('testData', 'Select the folder that contains the current
subject data');
 % subject folder directory
 ikdir=uigetdir(subjectdir, 'Select the folder that contains the IK data');
```

```
 % inverse kinematics results file folder directory
 grfdir=uigetdir(subjectdir, 'Select the folder that contains the .mot force
plate data');
 % ground reaction force
 [fileorig1 origpath1 index]=uigetfile('*.xml','select the setup force file');
 % inverse dynamics force setup file
 [fileorig2 origpath2 index]=uigetfile('*.xml','select the IK setup file');
 % inverse dynamics setup file
 ik=dir(fullfile(ikdir, '*ao*'));
 for i=1:length(ik)
     name=ik(i).name;
     aonum(i)=sscanf(name,'t%dao'); % scan for all the t*ao IK results files
 end
 [n,ind]=sort(aonum); % sort the IK results files to prevent mistakes such
 % that t10ao rank higher than t1ao
 grf=dir(fullfile(grfdir, '*.mot*')); % auto-find the ground reaction force
files with .mot
 % file extension
 % This section do gaussian filtering of the ground reaction force signal
 % and save it to a new .mot file under the IK results folder
 if length(n)==length(grf)
     for i=n
         trial=[grfdir '\' grf(i).name];
         file1=Osim.readMOT(trial);
         data1=table2array(file1);
         tv=data1(:,1);
         tstart(i)=tv(1);
         tend(i)=tv(end);
         for j=2:size(data1,2)
             %data1(:,j)=gaussfilt(tv,data1(:,j),0.01);
             fc=5;
             fs=1/0.0017;
             wn=fc*2/fs;
             [b,a]=butter(1,wn);
             data1(:,j)=filtfilt(b,a,data1(:,j));
         end
         vvx1=data1(:,2);
         vvy1=data1(:,3);
         vvz1=data1(:,4);
         vpx1=data1(:,5);
         vpy1=data1(:,6);
         vpz1=data1(:,7);
         vtx1=data1(:,8);
         vty1=data1(:,9);
         vtz1=data1(:,10);
         vvx2=data1(:,11);
```

```
        vvy2=data1(:,12);
        vvz2=data1(:,13);
        vpx2=data1(:,14);
        vpy2=data1(:,15);
        vpz2=data1(:,16);
        vtx2=data1(:,17);
        vty2=data1(:,18);
        vtz2=data1(:,19);
        time=tv;
        ground_force1_vx=-vvz1;
        ground_force1_vy=vvy1;
        ground_force1_vz=vvx1;
        ground_force1_px=-vpz1;
        ground_force1_py=vpy1;
        ground_force1_pz=vpx1;
        ground_force2_vx=-vvz2;
        ground_force2_vy=vvy2;
        ground_force2_vz=vvx2;
        ground_force2_px=-vpz2;
        ground_force2_py=vpy2;
        ground_force2_pz=vpx2;
        ground_torque1_x=-vtz1;
        ground_torque1_y=vty1;
        ground_torque1_z=vtx1;
        ground_torque2_x=-vtz2;
        ground_torque2_y=vty2;
        ground_torque2_z=vtx2;
        file3=table(time,ground_force1_vx,ground_force1_vy, ...
            ground_force1_vz,ground_force1_px,ground_force1_py, ...
            ground_force1_pz,ground_torque1_x,ground_torque1_y, ...
            ground_torque1_z,ground_force2_vx,ground_force2_vy, ...
            ground_force2_vz,ground_force2_px,ground_force2_py, ...
            ground_force2_pz,ground_torque2_x,ground_torque2_y, ...
            ground_torque2_z);
        grffile=[ikdir '\trial_' num2str(i) '_grf.mot'];
        grfstr(i)=string(grffile);
        mot=Osim.writeMOT(file3,'FilePath',grffile);
    end
else
    disp('the files number do not match');
end
% This section changes the preset force file and copy the changed force
% setup file to the IK results directory
for i=n
    idffile(i)=string([ikdir '\' sprintf('idf%d.xml',i)]);
    copyfile([origpath1 '\' fileorig1],idffile(i));
```

```
    S=fileread(idffile(i));
    % Replace preset strings with custom file name and directory:
    S=strrep(S, 'grf_file',grfstr(i));
    fname=fopen(idffile(i),'w');
    fwrite(fname, S);
    fclose(fname);
end
% This section changes the preset IK setup file and copy the changed
% setup file to the IK results directory
for i=n
    idsfile(i)=string([ikdir '\' sprintf('idsetting%d.xml',i)]);
    copyfile([origpath2 '\' fileorig2],idsfile(i));
    S=fileread(idsfile(i));
    % Replace preset strings with custom files' names and directories:
    S=strrep(S, 'subjectdir', subjectdir);
    S=strrep(S, 'sidf', idffile(i));
    S=strrep(S, 't.mot', [ikdir '\' sprintf('t%d.mot',i)]);
    S=strrep(S, 'id.sto', sprintf('id%d.sto',i));
    S=strrep(S, 'bf.sto', sprintf('bf%d.sto',i));
    S=strrep(S, 'start', num2str(tstart(i)));
    S=strrep(S, 'end', num2str(tend(i)));
    fname=fopen(idsfile(i),'w');
    fwrite(fname, S);
    fclose(fname);
end
% This section sets up the OpenSim environment and runs the inverse
% dynamics
import org.opensim.modeling.*
[mfile modelpath index]=uigetfile('*.osim','select the IK model file');
% Get the OpenSim model file for the inverse dynamics
for i=n
    sDir= ikdir; %Subject directory
    % Choose a generic setup file to work from
    Setup=idsfile(i);
    idTool = InverseDynamicsTool(Setup);
    % Load the model and initialize
    model = Model(mfile);
    model.initSystem();
    % Tell Tool to use the loaded model
    idTool.setModel(model);
    idTool.run();
end
```

Appendix C: Two Sample .XML Files for ID Setup in MATLAB

Appendix C.1: External Loads Setup


```
<?xml version="1.0" encoding="UTF-8" ?>
<OpenSimDocument Version="40000">
	<ExternalLoads name="externalloads">
		<objects>
			<ExternalForce name="right">
				<!--Name of the body the force is applied to.-->
				<applied_to_body>calcn_r</applied_to_body>
				<!--Name of the body the force is expressed in (default is ground).-
->
				<force_expressed_in_body>ground</force_expressed_in_body>
				<!--Name of the body the point is expressed in (default is ground).-
->
				<point_expressed_in_body>ground</point_expressed_in_body>
				<!--Identifier (string) to locate the force to be applied in the data
source.-->
				<force_identifier>ground_force1_v</force_identifier>
				<!--Identifier (string) to locate the point to be applied in the data
source.-->
				<point_identifier>ground_force1_p</point_identifier>
				<!--Identifier (string) to locate the torque to be applied in the data
source.-->
				<torque_identifier>ground_torque1_</torque_identifier>
				<!--Name of the data source (Storage) that will supply the force
data.-->
				<data_source_name>grf_file</data_source_name>
			</ExternalForce>
			<ExternalForce name="left">
				<!--Name of the body the force is applied to.-->
```

```
				<applied_to_body>calcn_l</applied_to_body>
				<!--Name of the body the force is expressed in (default is ground).-
->
				<force_expressed_in_body>ground</force_expressed_in_body>
				<!--Name of the body the point is expressed in (default is ground).-
->
				<point_expressed_in_body>ground</point_expressed_in_body>
				<!--Identifier (string) to locate the force to be applied in the data
source.-->
				<force_identifier>ground_force2_v</force_identifier>
				<!--Identifier (string) to locate the point to be applied in the data
source.-->
				<point_identifier>ground_force2_p</point_identifier>
				<!--Identifier (string) to locate the torque to be applied in the data
source.-->
				<torque_identifier>ground_torque2_</torque_identifier>
				<!--Name of the data source (Storage) that will supply the force
data.-->
				<data_source_name>grf_file</data_source_name>
			</ExternalForce>
		</objects>
		<groups />
		<!--Storage file (.sto) containing (3) components of force and/or torque and point
of application.Note: this file overrides the data source specified by the individual external forces
if specified.-->
		<datafile>grf_file</datafile>
	</ExternalLoads>
</OpenSimDocument>
```


Appendix C.2: ID Setup


```
<?xml version="1.0" encoding="UTF-8"?>
```

```xml
<OpenSimDocument Version="40000">
	<InverseDynamicsTool name="3DGaitModel2392-scaled-scaled-scaled">
		<!--Directory used for writing results.-->
		<results_directory> subjectdir\idresults </results_directory>
		<!--Directory for input files-->
		<input_directory>  </input_directory>
		<!--Name of the .osim file used to construct a model.-->
		<model_file> Unassigned </model_file>
		<!--Time range over which the inverse dynamics problem is solved.-->
		<time_range> start end </time_range>
		<!--List of forces by individual or grouping name (e.g. All, actuators,
		   muscles, ...) to be excluded when computing model dynamics.-->
		<forces_to_exclude> Muscles </forces_to_exclude>
		<!--XML file (.xml) containing the external loads applied to the model as
		   a set of ExternalForce(s).-->
		<external_loads_file> sidf </external_loads_file>
		<!--The name of the file containing coordinate data. Can be a motion
		   (.mot) or a states (.sto) file.-->
		<coordinates_file> t.mot </coordinates_file>
		<!--Low-pass cut-off frequency for filtering the coordinates_file data
		   (currently does not apply to states_file or speeds_file). A negative
		   value results in no filtering. The default value is -1.0, so no
		   filtering.-->
		<lowpass_cutoff_frequency_for_coordinates>       6.00000000
</lowpass_cutoff_frequency_for_coordinates>
		<!--Name of the storage file (.sto) to which the results should be
		   written.-->
		<output_gen_force_file>subjectdir\id.sto</output_gen_force_file>
```

<!--List of joints (keyword All, for all joints) to report body forces acting at the joint frame expressed in ground.-->

<joints_to_report_body_forces />

<!--Name of the storage file (.sto) to which the body forces at specified joints are written.-->

<output_body_forces_file>subjectdir\bf.sto</output_body_forces_file>

</InverseDynamicsTool>

</OpenSimDocument>