C3d2OpenSim Documentation

# Quick Start Guide

1. Download and unzip the btk-matlab\_64 binaries windows.
2. Get the motionProcessing source code from the Github repo. Either a) Clone repo to your local machine b) Download Zip folder of source and unpack
3. Put folders for both btk and motionProcessing in your matlab path.
4. Type ‘c3d2Opensim’ in matlab. If successfully installed, a selection tool will pop up instructing you to select a c3d file. Select a standard c3d gait trial to test, or use a test c3d file located in the ‘motionProcessing/tests/testData’ directory.
5. Edit the ‘Set Manual Values’ in the c3d2Opensim.m file convert files based on your own lab/data setup.

# C3D Readers

## btk and btk\_loadc3d

Btk (Arnaud Barre and Stephane Armand) are a binaries (mex functions) that matlab can use to read a c3d file. Since c3d's are not standardized across companies and/or labs, it is extremely difficult for c3d reading to be generic. BTK does a great job of reading, marker, forceplate, analog and event data and outputing it.

btk\_loadc3d() (Glen Litchwark) has written a very nice interface function for the btk binaries that stores data given from btk into a particular structure for analysis. Marker, forceplate, analoge and event data can all be stored in a consistent way that allows for easy, consistent manipulation and analysis.

# Function Documentation

## rotateCoordinateSys()

Assumed orientation in OpenSim = X (long Axis, forward/back)

Y (up)

Z (90 degrees to x-y plane)

We assume that X is the long axis of your lab and Y is up. For most labs

who have long axis X and Z up, this only requires a 90 degree rotation about X.

If your lab is in some other orientation, you will just need to do 2 rotations to put your data in the aforementioned orientation

'one rotation about x'

rotation.axis = {'x'};

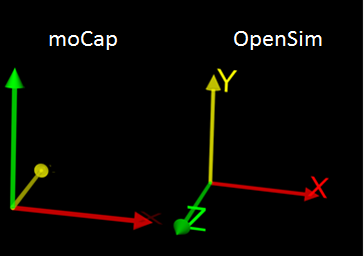
rotation.value= [90];

'two rotations when the lab frame has Y as the long axis and Z is up'

rotation.axis = {'z' 'x'};

rotation.value= [90 90];

The typical mocap XYZ global frame configuration has ‘Y’ being up while OpenSim has ‘Z’ up. This 90-degree rotation about X will need to be applied to the marker and force data.



## filterData()

## forces2Global()

## grfProcessing()

## copCalc()

## connectBody2Forces()

## PrintMot()

## PrintTRC()