C3D (.c3d) Files

The typical file format for experimental motion capture data in the biomechanics community is C3D. C3D is a flexible format that can store marker, force plate, EMG, and event data. Despite being very flexible, data in C3D files are stored in binary format and require specialized readers to access. To read more information on the C3D file format, we encourage you to visit C3D.org to learn more.

Starting with OpenSim 4.0, C3D reading and conversion into OpenSim formats is available. Currently, use of C3D reading is limited to C++ and scripting and we will be working to support direct C3D reading in the GUI in future releases. The easiest way to use OpenSim to read your C3D data is through the Matlab interface. Once you have setup OpenSim use in Matlab, you can read C3D files and write marker and force data to .trc and .mot file formats, easily.

Starting with OpenSim 4.2, OpenSim defaults to using ezc3d (https://github.com/pyomeca/ezc3d) to parse c3d files.

⚠ Prior to OpenSim version 4.2, C3DFileAdapter only supported reading Type-2 Force plates (AMTI & Bertec) and not Type-3 (Kistler), however this limitation has been lifted with the use of ezC3D in version 4.2 and moving forward. Please contact the support of ezc3d project above if you run into issues using it.

- Reading C3D files through Matlab
- Reading C3D files through Python and C++
- Importing Force Data: COP and NaNs

Reading C3D files through Matlab

osimC3D

We have included a utility in Matlab that can be used to perform some common operations, such as rotating data, converting data into Matlab data types, and writing marker and force data to OpenSim file formats. The Matlab file can be found in your resources directory **/Code/Matlab/Utilities/osimC3D.m**. An example of using the osimC3D function is in the expandable section below. In this example, we

- read a C3D file containing markers and forces,
- get some information about the data (rate, number of markers, and number of forces),
- rotate the data, and
- then write the markers to a .trc file, and the forces to a .mot file.

Unknown macro: 'view-git-file'

Reading C3D files through Python and C++

Some example code for using C3DFileAdapter through **Python** can be found below. This code is part of a test script that runs the C3D reader and writes data marker and force data to a storage (.sto) file.

```
tables = C3DFileAdapter.readFile(os.path.join(test_dir, 'walking2.c3d'), 1)
markers = tables['markers']
forces = tables['forces']

# Marker data read from C3D.
markers = tables['markers']

# Flatten marker data.
markersFlat = markers.flatten()

# Make sure flattenned marker data is writable/readable to/from file.
markersFilename = 'markers.sto'
stoAdapter = osim.STOFileAdapter()
stoAdapter.write(markersFlat, markersFilename)
markersDouble = stoAdapter.read(markersFilename)

# Forces data read from C3d.
forces = tables['forces']
fpCalMats = forces.getTableMetaDataVectorMatrix("CalibrationMatrices")
fpCorners = forces.getTableMetaDataVectorMatrix("Corners")
fpOrigins = forces.getTableMetaDataVectorMatrix("Origins")

# Flatten forces data.
forcesFlat = forces.flatten()

# Make sure flattenned forces data is writable/readable to/from file.
forcesFilename = 'forces.sto'
stoAdapter.write(forcesFlat, forcesFilename)
forcesDouble = stoAdapter.read(forcesFilename)

# Clean up.
os.remove(markersFilename)
os.remove(forcesFilename)
```

Example code for using the C3DFileAdapter in **C++** is found below. This code is part of a test script that runs the C3D reader and checks for correct values.

OpenSim 4.0

Unknown macro: 'view-git-file'

OpenSim 4.1

Unknown macro: 'view-git-file'

Unknown macro: 'view-git-file'

Importing Force Data: COP and NaNs

When you import your force data, you can use the origin of the force plate or center of pressure (COP) to express the forces. The C3DFileAdapter includes an input parameter (ForceLocation) that allows you to choose OriginOfForcePlate or CenterOfPressure. Using the COP is helpful for visualization, but can lead to NaN values. In particular, when the force values go to zero, the COP value is undefined (NaN). This is problematic during Inverse Dynamics analysis as you will get all NaN values for the output moments. This is because OpenSim splines the force and COP data before computing joint moments and the splines are undefined in the presence of NaNs for your input data. If you find that your mot files are containing NaNs for COPs after conversion, you can use the OriginOfForcePlate flag when you import your C3D data.

Next: Tools for Preparing Motion Data

Previous: Storage (.sto) Files

Home: Preparing Your Data