**SETTING UP OPENSIM PIPELINE – BFUNC**

**BFUNC pipeline for inverse kinematics and dynamics through OpenSim and Python.**

**These instructions are based on using Spyder and Anaconda Prompt for getting setup.**

**Written by Emily Eichenlaub 02/13/2025**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**STEP 1 – Connect Python and OpenSim**

Set up Python to OpenSim API

<https://simtk-confluence.stanford.edu:8443/display/OpenSim/Scripting+in+Python>

**STEP 2 – Create virtual environment & install necessary packages**

1. Download [C3D viewer](https://github.com/pyomeca/pyomeca)
2. Within Anaconda Prompt, navigate to downloaded PyOSim folder

conda cd <path>

1. Create virtual environment and install the necessary packages from requirements.txt file (downloaded in folder)

conda create --name <env\_name> --file requirements.txt

1. Activate environment
   1. **You will need to activate your environment each time you start a new session to run the pipeline**

conda activate <env\_name>

**NECESSARY PACKAGES:**

numpy

pandas

scipy.spatial.transform

scipy.linalg

cipy.signal

sys

opensim as osim

json

matplotlib.pyplot as plt

**STEP 3 – Data Setup**

**Folder structure**

1. Once all data is cleaned, place individual subject folders within the same study folder
2. Each subject folder will contain clean .c3d files

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**STEP 4 – Pipeline**

**Codes:**

1. MainOSimSetup.py – RUN FIRST, main setup code to convert data to be OpenSim compatible.

|  |  |
| --- | --- |
| Input | Subject folders with .c3d files |
| Output | Subjects.json   * Mass structure that houses information about each subject & all trials * Saved in study folder |

1. OSimSetupFunctions.py – functions called by MainOSimSetup.py

|  |  |
| --- | --- |
| **Function** | **Task** |
| C3D2OpenSim | converts .c3d files to OpenSim compatible files (.trc and .mot) & detects gait events |
| writeTRC | writes TRC files from .c3d marker data |
| LoadGRF | calculates subject mass from static trial |
| extract\_strength\_data | extracts strength data (5 torques at peak flexion/extension) |

1. MainOSim.py – RUN SECOND, main code to run OpenSim Tasks (scaling, IK, ID)

|  |  |
| --- | --- |
| Input | .trc and .mot files (created during setup)  Subjects.json file (created during setup) |
| Output | Scales generic osim model for each subject  Inverse Kinematics  Inverse Dynamics |

1. OSimProcFunctions.py – functions called by MainOSim.py

|  |  |
| --- | --- |
| **Function** | **Task** |
| Scale | Scales generic osim model for each subject based on static trial |
| readTRC | Reads TRC files |
| writeTRC | Writes and saves TRC files |
| MakeVirtualMkr | Creates virtual markers used for scaling purposes. |
| IK | Calls in subject specific model, setup files, and mocap data. Iterates through each trial to run IK in OpenSim. Saves output in subject folder (i.e., |
| ID | Calls in subject specific model, setup files, and GRF/IK data. Iterates through each trial to run ID in OpenSim. Saves output in subject folder |

**RESEARCHER/SITE SPECIFIC CHANGES**

1. Update file paths for your data/where you store the pipeline

MainOSimSetup: lines 35, 36

* + Note: pipeline is currently setup for strength and mocap data to be in separate folders.

MainOSim: lines 25, 31, 41, 44, 63

1. Change site in settings

MainOSimSetup: line 40

MainOSim: line 34