Tutorial 2 – Data Preprocessing

The NMSM Pipeline expects input data to be in a specific format for the tools to work properly. The data preprocessing sub-tool focuses on processing raw kinematic, force, and EMG data into a usable format. Data preprocessing is the next step in the pipeline after JMP, and is required to proceed with MTP, NCP, GCP, and finally, Treatment Optimization.

The inputs to data preprocessing are experimental IK motions, ID loads, ground reaction forces and moments, geometric muscle data, and EMG data. The tool then crops and filters the relevant quantities for the NMSM Pipeline tools to run.

Data preparation:

1. Open the OpenSim model **UF_Subject_3_reduced_muscles.osim** in the OpenSim GUI.
2. Run the *Inverse Kinematics tool* on the model using the settings file
   input_data\IKSettings.xml
   a. This settings file runs IK using 4 seconds of marker data for the gait trial to be analyzed. All weights are set to 1 for simplicity.
3. Run the *Inverse Dynamics tool* on the model using the settings file
   input_data\IDSettings.xml
   a. This settings file runs ID using the previous filtered IK results, and external force data in electrical center format.
4. Run the *Muscle Analysis tool* on the model using the settings file
   input_data\MASettings.xml
   a. This settings file runs MA using the previous IK results to calculate muscle-tendon lengths and moment arms that will be used later in the pipeline.

Data Preprocessing:

1. Open the MATLAB script **preprocessing.m** and examine the file.
2. This matlab script does 4 primary tasks:
   a. **Process EMG data**: Starting from the raw EMG file, the script high pass filters, demeans, rectifies, and low pass filters the EMG signals. Next, any remaining negative EMG values are set to zero. EMG signals are then offset so that the minimum value of each signal is 0. Finally, EMG signals are normalized so that the max value of each signal is 1.
   b. **Create muscle-tendon velocities**: The script low pass filters the muscle-tendon lengths created by Muscle Analysis, splines the filtered data using GCV splines, and then differentiates the filtered data.
   c. **Crop data:** A user input to preprocessing is a set of time pairs for which the data should be cropped to. One file is created for each time pair specified by the variable **trialTimePairs**.

d. **Filter data:** All data are low pass filtered using the cutoff frequency specified by **inputSettings.cutoffFrequency**.
3. Processed data are output into the directory **preprocessed**.
4. Before moving onto the next tools, it is important to visualize your preprocessed data to make sure it looks correct. Run **PlotPreprocessed.m** to do this.