**Folder: Human Data**

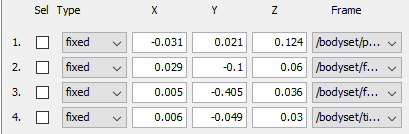
The most important files in this folder are BiMuscleData.m and MonoMuscleData.m which are class constructors for calculating values around human muscles.

**Files and Descriptions:**

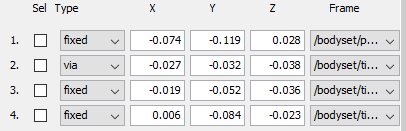
**BiMuscleData.m** – This class calculates parameters for biarticular human muscles.

Inputs:

* Name – string that represents the name of the muscle
* Location – the location points as listed in OpenSim for that specific muscle. The location variable is an nx3 matrix, where every row is the next point of the muscle segment. The columns are the x, y, and z location of that muscle point. The matrix should be in the same orientation and layout as the points listed in Figure 1.
* Cross – variable that represents which row in location corresponds to a crossing of a joint or a change in reference frame. When looking at the points on OpenSim, this is where the frame changes. In figure 1, the cross should be equal to [2, 4] as the first point to cross a joint is point 2, and the second point to cross a joint is point 4. In Figure 2, the cross should be equal to [2, 2] as the second point spans the hip joint AND the knee joint.

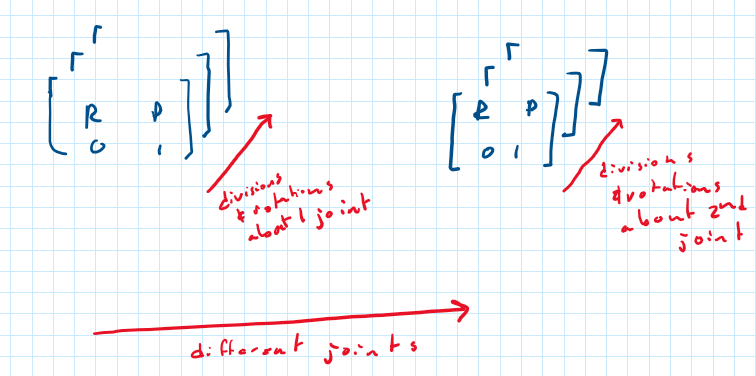


**Figure 1.** “Geometry Path” for a biarticular muscle (TFL) in which two different segments each span their own joint.



**Figure 2.** “Geometry Path” for a biarticular muscle (grac) in which a single muscle segment spans two joints.

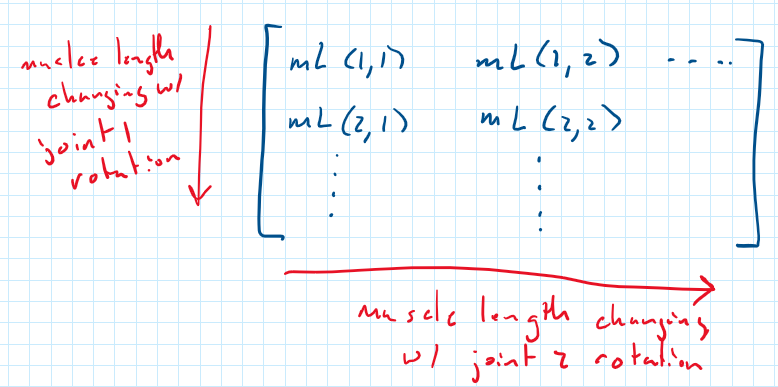
* Mif – the maximum isometric muscle force
* Tsl – the tendon slack length
* Pennation – the pennation angle of the muscle
* Ofl – the optimum fiber length of the muscle
* T – the transformation tensor for the joints that the muscle spans. The transformation tensor is a 4x4xnx2 matrix (i x j x k x L). The i and j elements represent the transformation matrix for a joint in an orientation, k represents a new orientation of that same joint, and L is the different joints that the muscle crosses over.



**Figure 3.** Visual representation of the transformation tensors

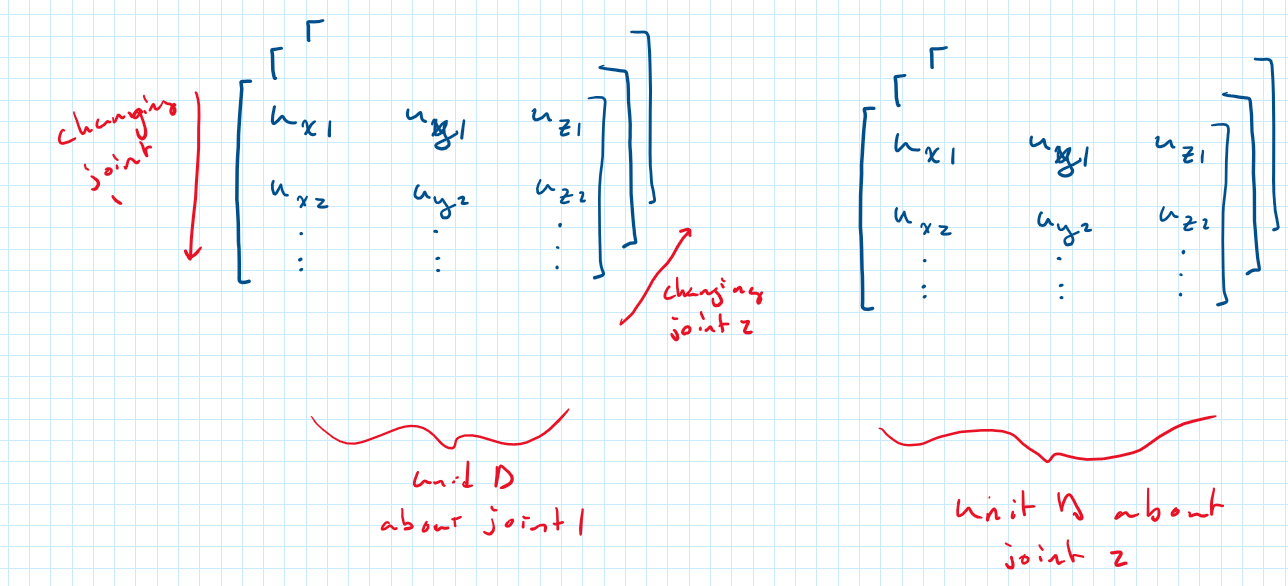
Outputs:

* MuscleLength – the total length of the muscle in every orientation of the joints. An n x m matrix. n elements represent the rotating the first joint while the second joint remains stationary and m elements represent rotating the second joint while the first joint remains stationary.



**Figure 4.** Visual representation of the MuscleLength variable

* UnitDirection – the unit direction of the line of action that the force acts along about a joint. This unit direction is an n x 3 x m x 2 matrix. n elements represent changing the first joint, columns represent the x, y, and z components of the unit vector, m represents changing the second joint, and the 4th element represents the unit vector about either the first or second joint.



**Figure 5.** Visual representation of the UnitDirection variable

* Force – the maximum isometric force that a muscle can produce in every given configuration. It is calculated by using an equilibrium algorithm derived by Hoy 1990 and listed in Thelen 2003 and Millard 2013. The algorithm seeks a solution in which the contractile force generated by the muscle equals the force generated by the tendon, given an overall musculotendon length (provided by MuscleLength). The individual tendon and muscle lengths are also calculated through this algorithm. The variable “Force” is the product of the solution that is generated multipled by “UnitDirection”
* MomentArm – The vector moment arm, starting at a joint center and extending perpendicularly to the line of action of the force that crosses over it. Derivations of it’s calculation can be found in the latex document “Calculating Moment Arms for Human Leg Muscles”. Structure of this tensor is the same as “UnitDirection”
* Torque – The torque that the muscle is able to create about each joint. The cross product between “MomentArm” and “Force”. The structure is the same as “UnitDirection”

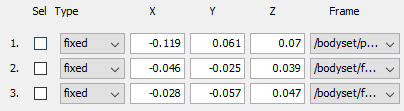
**HumanMuscleCalculation.m** – This script contains all of the locations and other variables necessary for the muscle classes to calculate. The entire script can be ran to calculate all muscle values or specific muscle properties can be copied and pasted into optimization scripts. It also includes transformation matrices for the hip and knee joint.

**KneeFunction.m** – This script is a demonstration of the how the x and y coordinates of the knee’s center of rotation changes and the knee actuates.

**MonoMuscleData.m** – This class calculates parameters for monoarticular human muscles. The variables are the same as in BiMuscleData.m, except one dimension smaller.

Inputs:

* Name – string that represents the name of the muscle
* Location – the location points as listed in OpenSim for that specific muscle. The location variable is an nx3 matrix, where every row is the next point of the muscle segment. The columns are the x, y, and z location of that muscle point. The matrix should be in the same orientation and layout as the points listed in Figure 1.
* Cross – variable that represents which row in location corresponds to a crossing of a joint or a change in reference frame. When looking at the points on OpenSim, this is where the frame changes. This should only be a single scalar value, as a monoarticular muscle will only cross over one joint. In figure 6, cross will be equal to 2.



**Figure 6.** “Geometry Path” for a monoarticular muscle

* Mif – the maximum isometric muscle force
* Tsl – the tendon slack length
* Pennation – the pennation angle of the muscle
* Ofl – the optimum fiber length of the muscle
* T – the transformation tensor for the joints that the muscle spans. The transformation tensor is a 4x4xn matrix (i x j x k). The i and j elements represent the transformation matrix for a joint in an orientation, k represents a new orientation of that same joint.

Outputs:

* MuscleLength – the total length of the muscle in every orientation of the joints. An n x 1 vector. n elements represent the muscle length as the joint rotates.
* UnitDirection – the unit direction of the line of action that the force acts along about a joint. This unit direction is an n x 3 matrix. n elements represent rotating the joint and columns represent the x, y, and z components of the unit vector.
* Force – the maximum isometric force that a muscle can produce in every given configuration. It is calculated by using an equilibrium algorithm derived by Hoy 1990 and listed in Thelen 2003 and Millard 2013. The algorithm seeks a solution in which the contractile force generated by the muscle equals the force generated by the tendon, given an overall musculotendon length (provided by MuscleLength). The individual tendon and muscle lengths are also calculated through this algorithm. The variable “Force” is the product of the solution that is generated multipled by “UnitDirection”
* MomentArm – The vector moment arm, starting at a joint center and extending perpendicularly to the line of action of the force that crosses over it. Derivations of it’s calculation can be found in the latex document “Calculating Moment Arms for Human Leg Muscles”. Structure of this tensor is the same as “UnitDirection”
* Torque – The torque that the muscle is able to create about each joint. The cross product between “MomentArm” and “Force”. The structure is the same as “UnitDirection”