**Folder: Robot Data**

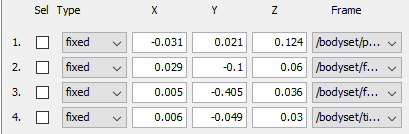
The most important files in this folder are BiPamData.m and MonoPamData.m which are class constructors for calculating values around PAMs for the robots.

**Files and Descriptions:**

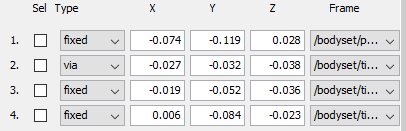
**BiPamData.m** – This class calculates parameters for biarticular Pams.

Inputs:

* Name – string that represents the name of the PAM or the muscle that the PAM will replace
* Location – generally taken from 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.

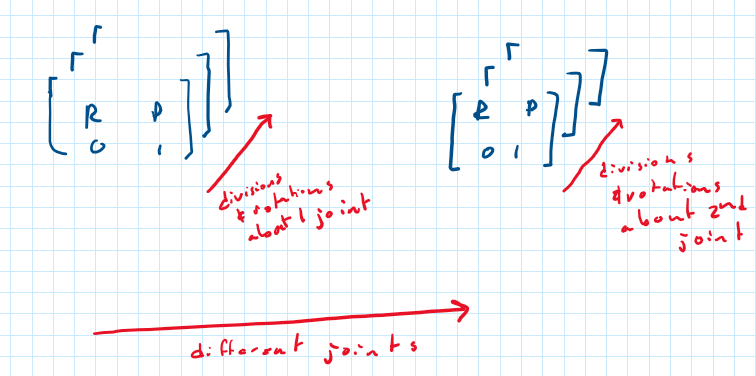


**Figure 1.** “Geometry Path” for a biarticular muscle (TFL) in which two different segments each span their own joint.



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

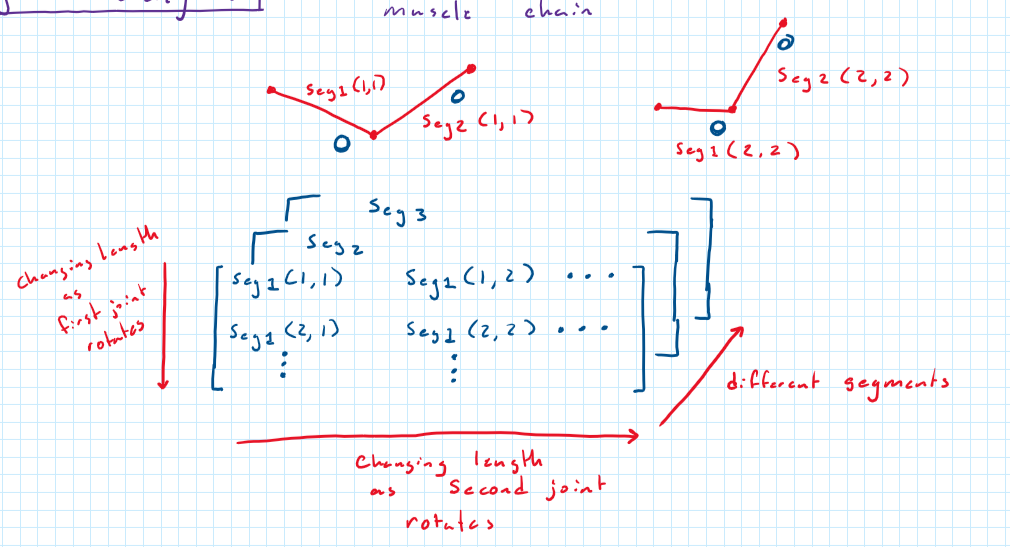
* Diameter – The proposed diameter for the PAM of interest. Choices are 10, 20, and 40.
* T – the transformation tensor for the joints that the PAM spans. The transformation tensor is a 4x4xnx2 matrix (i x j x k x L). The i and j elements represent the transformation matrix for a joint in an orientation, 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

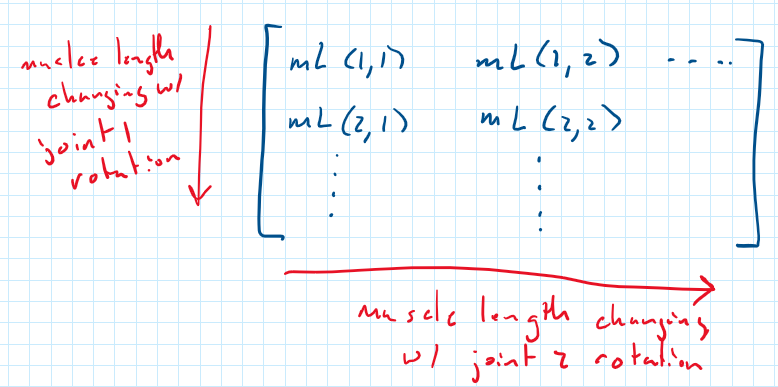
Outputs:

* SegmentLengths – length of every segment of a pam, defined by two points in location. Used for further analysis and outputs. Variable is a n x m x k matrix. Rows correspond with changing the first joint. Columns correspond with changing the second joint. The kth dimension is for different segments. Each element is a length measurement.



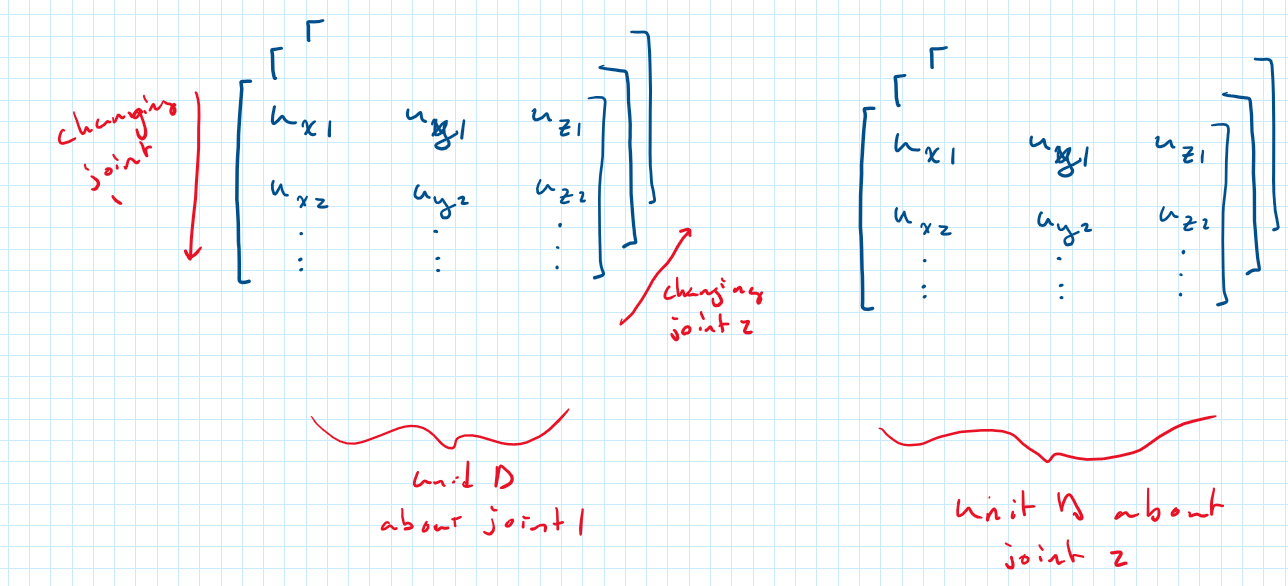
**Figure 4.** Visual representation of the transformation tensors

* LongestSegment – Matrix of the lengths of the segment that is the longest on average through different joint positions. Determined from averaging the individual segments from “SegmentLengths”. This segment will contain the PAM. Variable is an n x m matrix, with rows corresponding to revolving the first joint and columns correspond to revolving the second joint. Every element is a length measurement. Essentially this variable is one of the k elements from the variable “SegmentLengths”.
* MuscleLength – total length of the PAM and the tendon length. Lengths are listed for every joint orientation.



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

* RestingL – scalar value for the length of the PAM required.
* Contraction – The amount that the PAM will need to contract at every joint position. At 0% the PAM is not contracted at all. At 10%, the PAM is at 90% of its resting length value.
* LengthCheck – Boolean variable, either “Usable” or “Unusable”, determined by checking if the PAM must contract passed what is physically possible. Contraction conditions are based off assumptions from Festo and can/should be updated when more empirical evidence Is obtained.
* 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 6.** 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”

**MonoPamData.m** – This class calculates parameters for monoarticular PAMs. The variables are the same as in BiPamData.m, except one dimension smaller.

**MonoPamDataPhysicalExtensor.m** – This class was used for Lindie and Ben’s physical knee test experiment. It is the MonoPamData.m constructor, however it hard codes the length of the PAM, tendon, and air fittings, instead of coming up with a solution on its own. Specifically used for the knee extensor.

**MonoPamDataPhysicalFlexor.m** – This class was used for Lindie and Ben’s physical knee test experiment. It is the MonoPamData.m constructor, however it hard codes the length of the PAM, tendon, and air fittings, instead of coming up with a solution on its own. Specifically used for the knee flexor.

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

**MonoPamDataPinnedExtensor.m** – Same as MonoPamPhysicalExtensor constructor but for the pinned knee joint.

**MonoPamDataPhysicalFlexor.m** – Same as MonoPamDataPinnedExtensor.m but for a Flexor muscle.

**MonoPamDataExplicit.m** - Will calculate everything given all the inputs:

name, location, cross, diameter, t, rest, kmax, tendon, fit, pres

If only given the first 7 inputs its will assume tendon length = 0, pressure = 620 kPa, and fitting length = 0.0254 m.

**MonoPamDataExplicit\_balance.m** - Same as MonoPamDataExplicit but calculates a force balance between BPA and an artificial tendon. Not as up to date as MonoPamDataExplicit. Could be useful.

**MonoPamDataExplicit\_compare.m** - Same as MonoPamDataExplicit but calculates force using different methods. Not as up to date as MonoPamDataExplicit. Can probably delete.

Can probably delete MonoPamData[PinnedFlexor,PinnedExtensor,PhysicalFlexor,PhysicalExtensor]