**Folder: Functions**

This folder contains functions to be used in other scripts.

**Files and Descriptions:**

**costFunction** – This function serves as the primary cost function or objective function that the optimization code will be attempting to minimize throughout iterations. The first component of the cost output is the summed and normalized difference between the human torque and PAM replacement torque. The second component of the cost output is the weighted angle between the human torque directional vector and the PAM replacement torque directional vector.

Inputs: This function takes the human muscle torque and the PAM replacement torque as inputs.

Outputs: The output of this function is a C or cost value. The value is unitless and can be scaled to any arbitrary amount through the use of the gains Gt and Ga.

**RowVecTrans.m** – This function orients a vector in one reference frame into a second reference frame. Useful for bringing muscle location points in the first reference frame into a second reference frame.

Inputs:

* T – The transformation matrix that describes the orientation of the second reference frame with respect to the first reference frame.
* p – A row vector in the first reference frame

Outputs:

* v – The row vector, p, in the second reference frame

**RpToTrans** – This function comes from a robotics text book. It creates a transformation matrix from a rotation matrix and a distance column vector.

Inputs:

* R – The rotation matrix that describes the rotation of the second reference frame with respect to the first reference frame
* p – the column vector that describes the distance of the second reference frame origin from the first reference frame origin

Output:

* T – Transformation matrix which is the combination of R and p

**festo** – This function comes from Dr. Hunt’s work on characterizing the 10 mm I.D. Festo BPA. It calculates force based on current muscle length and resting length. It does interpolation of ForceStrainTable.mat from Dr. Hunt, assuming 620 kPa. For larger muscle diameters the force is scaled based off of the maximum theoretical force specified by festo.

Inputs:

* Lmt – Total Musculo-tendon length
* rest – resting BPA length
* dia – BPA diameter
* long – longest musculotendon length

Output:

* F – Force at maximum pressure

**festo2** – This function comes from Dr. Hunt’s work on characterizing the 10 mm I.D. Festo BPA. It calculates force based on current muscle length and resting length. It does interpolation of ForceStrainTable.mat from Dr. Hunt, but wants to know pressure (col). For larger muscle diameters the force is scaled based off of the maximum theoretical force specified by festo.

Inputs:

* Lmt – Total Musculo-tendon length
* rest – resting BPA length
* dia – BPA diameter
* long – longest musculotendon length
* col – column of forcestrain table.

Output:

* F – Force at specified pressure

**festo3** – This function comes from Dr. Hunt’s work on characterizing the 10 mm I.D. Festo BPA. It calculates force based on current muscle length, resting length, maximum contracted length, pressure. It takes into account if there are fittings or artificial tendons attached to the musculotendon length. It does interpolation of ForceStrainTable.mat from Dr. Hunt for relative strain between 0 and 1. For stretched lengths up to 3% elongation, data from the Festo muscleSIM tool is used. Larger diameter muscles are scaled based off of Festo’s maximum force.

Inputs:

%Lmt == muscle-tendon length, scalar

%rest == resting length of artificial muscle, "size" from Size function

%dia == diameter of Festo tube, from Size function

%long == longest musculotendon length

%pres == pressure in kPa

%kmax == maximum measured contraction length. Input as length, will

%convert to percent in the code.

%ten == tendon length

%Fitting == fitting length

Output:

* F – Force at pressure and length

**festo4** – This function is designed to calculate force in 20 and 40 mm BPAs using data from Festo. The data were obtained in Festo’s muscleSIM tool.

Inputs:

* dia – BPA diameter (mm)
* pres – pressure in actuator (kPa)
* contract – percent contraction

Output:

* F – Force at specified pressure

**FestoLookup** – This script creates fits for 20mm and 40mm Festo BPA data obtained using Festo’s muscleSIM tool.

Inputs:

* none

Output:

* f40 – Force equation for 40mm Festo BPA given contraction and pressure.
* f20 – Force equation for 20mm Festo BPA given contraction and pressure.

**HuntEq** – This function is Dr. Hunt’s equation from his paper, “Modeling Length Effects of Braided Pneumatic Actuators.”

Inputs:

* epsilon = contraction of BPA
* epsilon\_max =maximum contraction of BPA at 620 kPa.
* P = BPA pressure.
* S = inflating, deflating, or isometric condition of BPA.

Output:

* Force, N

**LookupTable\_generate** – This script attempts to recreate the data in the ForceStrainTable using HuntEq.m.

Inputs:

* none, but it is possible to vary epsilon\_max at the beginning of the script.

Output:

* B matrices for condions S = 0, S = 1, and S = -1 in the same format as ForceStrainTable.
* A matrices are B matrices with first column (relative strain) removed and all excess 0 values removed, to improve fit function.

**f10** – function to obtain force given relative strain and pressure in 10 mm BPA. parameters were derived using ForceStrainForFit.mat, which is the same as ForceStrainTable.mat but without the column for relative strain and only one zero value in each remaining column.

Inputs:

* x – relative strain
* y – Pressure (kPa)

Output:

* F = Force, N