Bipedal Robot Legs Project

by

Ben Bolen

A project submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Mechanical Engineering

Thesis Committee:
Alex Hunt, Chair
Marek Perkowski

Portland State University
2019

Table of contents

List of Figures

List of Tables

Nomenclature

Biomechanics; biomimetics; bipedal robot; PAMs; moment arm; joint torque; robotics; humanoid; biarticular muscles; uniarticular muscles; over-actuated system; OpenSim; Matlab

Abstract

Bipedal robotic leg kinematics are a key component in designing biomimetic humanoid robots. This work describes the process of designing a bipedal robot utilizing Pneumatic Artificial Muscles. Pneumatic Artificial Muscles (PAMs) offer similar force and activation times to real muscles, while being lightweight and not demanding too much power. Muscle length, moment arm, and force are analyzed for different joint configurations, based on muscle attachment and wrapping points. These parameters are then used to find joint torques for muscles. Joint torques for individual muscles and groups of muscles are compared to existing models. Range of motion is reduced to remain within PAM strain limits. Matlab tools created for this project will facilitate further design refinement.

1. Introduction

Biomimetic robots attempt to replicate the anatomical motions and biological systems of living organisms. Of particular interest to humans are biomimetic bipedal humanoid robots. One approach in robot design is the task of engineering robotic systems to match observed biological principles, and another is to use biomimetic robots to gain insight into the inner workings of biological systems.

Bipedal jumping [1], hopping [2], walking [3-6], and swimming [7] robots have been designed. The functionality of these robots is usually limited to the tasks and applications they are designed for. Other robots have been designed that are based more on the human musculoskeletal structure [8] [9] with the goal of making them more flexible in the tasks they can carry out. Designing a humanoid robot based on a high fidelity to human systems can improve understanding of internal human systems and human biomechanics [6][10].

A traditional robot design is to use rotary actuators that directly drive joints. A biomimetic design seeks to use linear actuators that are analogous to human muscles. McKibben Pneumatic Artificial Muscles (PAMs) are very similar to the static mechanical features of biological muscles [11]. Force-length properties of McKibben pneumatic actuators have been shown in experiments to be similar to muscles [12]. PAMs have a power-to-volume and power-to-weight ratios equal to or greater than other actuators used in robotics, and they can be used in accordance with the Hill muscle model [13].

Having a tool that can accurately simulate and analyze Hill-type muscle models is an important step in comparing and evaluating any biomimetic robot humanoid designs.

OpenSim is a freely available, open-source software platform for the biomechanics community to perform musculoskeletal modeling and simulations for *in silico* investigations [14]. The editable musculoskeletal modeling and model visualization provided by OpenSim can help in the design of an artificial musculo-skeletal system for biomimetic robots.

This aim of this work is to create the tools and lay the foundations of the design of a high fidelity biomimetic bipedal humanoid robot using Festo PAMs. The tools created first accurately reproduced the OpenSim methodology and model specific parameters of the benchmark human model Gait2392 in Matlab. Then a similar tool was created for the robot design, modified to use Festo fluidic muscles. The robot model also removed certain muscle actuators and changed attachment points to accommodate experimentally determined maximum contraction limits. Individual muscles are compared as are groups of muscles based on their anatomical function [15, 16]. Next steps in the bipedal humanoid development process are described.

2. Materials and Methods

2.1 Bone Segments

For this project, the model Gait 2392 was used from OpenSim. Gait 2392 has 23 degrees of freedom, 92 musculotendon actuators to represent 76 real muscles of torso and lower limbs, and 12 rigid-body segments. This model is based on a subject that is about 1.8 m tall [17]. Shank and foot bone data are obtained from Stredney et al. [18].

Joint centers are placed in the same location as OpenSim (see Fig. 1 and Table 1). Using the same bone segment geometry in the robot body and OpenSim eliminates the need to perform scaling. Knee origin uses same function as OpenSim (Fig. 2). Patella is removed, and the quadriceps insertion on the patella is modeled in the tibia frame as a function of knee angle.

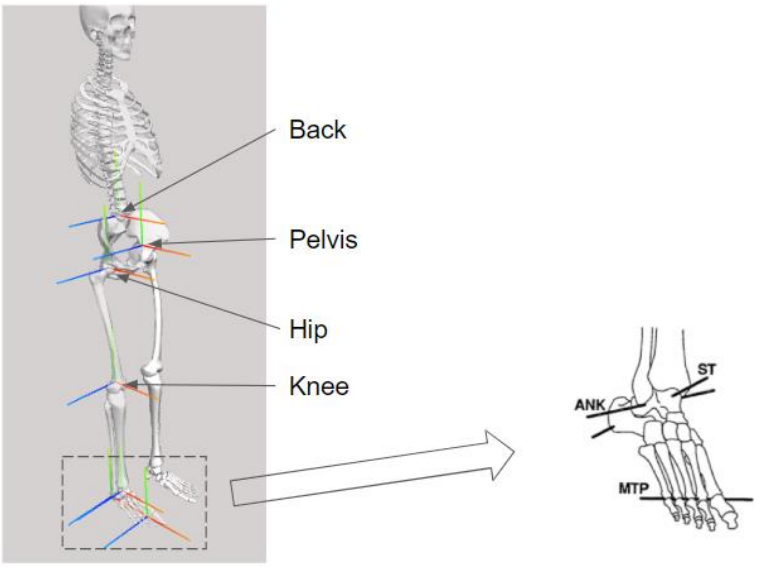


Fig. 1. Joint locations and names. Close-up of ankle from Delp, et al [19]. ANK is the ankle joint, ST is the subtalar joint, and MTP is the metatarsophalangeal joint.

Table 1: Joint locations. Knee position is a function of knee angle θ_K.

Segment		**Origin (m)**		**Parent Frame**
	X	***Y***	***Z***	
Back	-0.1007	0.0815	0	w.r.t. Pelvis
Pelvis	0	0	0	
Hip	-0.707	-0.0661	0.0835	w.r.t. Pelvis
Knee	fcn1(θ_K)	fcn2(θ_K)	0	w.r.t. Hip
Ankle	0	-0.43	0	w.r.t. Knee
Subtalar	-0.04877	-0.04195	0.00792	w.r.t. Ankle
MTP	0.1788	-0.002	0.00108	w.r.t. Subtalar

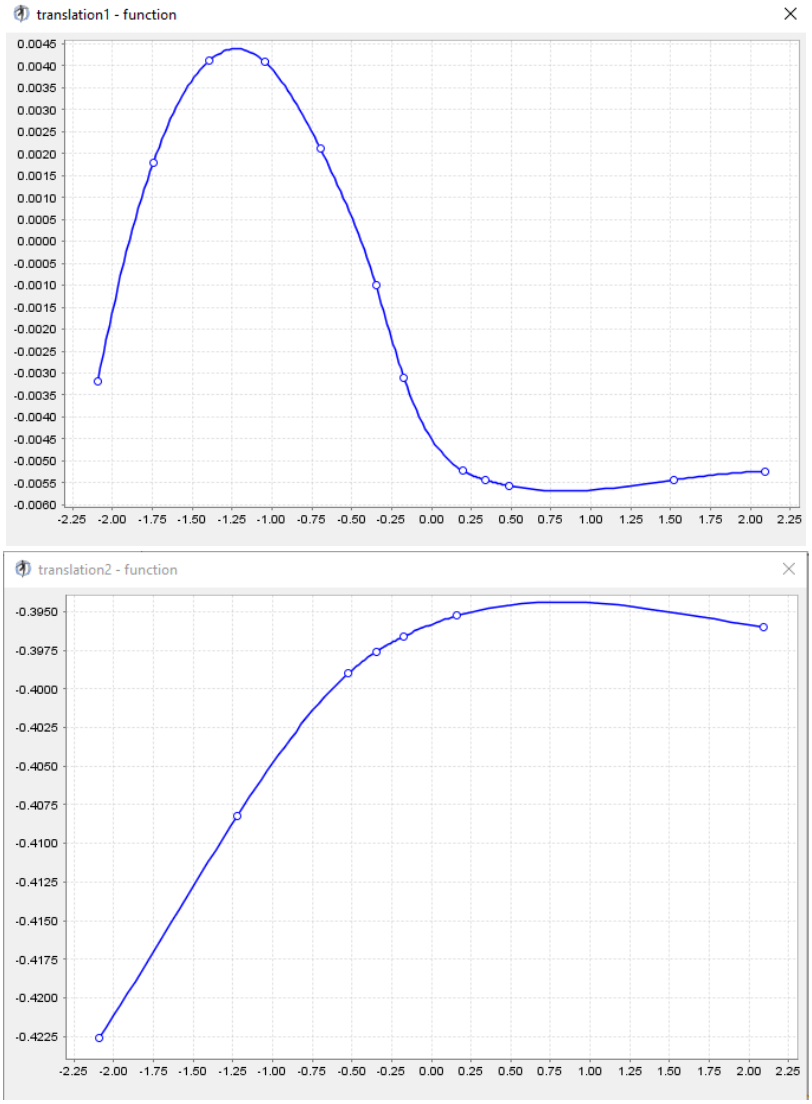


Fig. 2. Knee position (m) as a function of knee angle (rad) for X coordinate (top frame) and Y coordinate (bottom frame).

Table 2: Axis of revolution for selected joints

Segment		**Axis of Rotation ($\omega = \omega_1, \omega_2, \omega_3$)**	
	X	***Y***	***Z***
Knee	0	0	1
Ankle	-0.10501355	-0.17402245	0.97912632
Subtalar	0.7871796	0.60474746	-0.12094949
MTP	-0.5809544	0	0.81393611

The rotation matrices R for the ball and socket joints (i.e. Hip and Back) are:

R = Rot($\underline{x}$, θ)* Rot($\underline{y}$, θ)* Rot($\underline{z}$, θ)

where

$$\mathrm{Rot}(\hat{x},\theta)=\begin{bmatrix}1&0&0\\0&\cos\theta&-\sin\theta\\0&\sin\theta&\cos\theta\end{bmatrix}\quad \mathrm{Rot}(\hat{y},\theta)=\begin{bmatrix}\cos\theta&0&\sin\theta\\0&1&0\\-\sin\theta&0&\cos\theta\end{bmatrix}\quad \mathrm{Rot}(\hat{z},\theta)=\begin{bmatrix}\cos\theta&-\sin\theta&0\\\sin\theta&\cos\theta&0\\0&0&1\end{bmatrix}$$

The rotation matrix for the knee is Rot($\underline{z}$, θ_k), and the rotation matrix for each the three revolute foot joints is

$$\mathrm{Rot}(\hat{\omega},\theta)=$$
$$\begin{bmatrix}c_\theta+\hat{\omega}_1^2(1-c_\theta)&\hat{\omega}_1\hat{\omega}_2(1-c_\theta)-\hat{\omega}_3 s_\theta&\hat{\omega}_1\hat{\omega}_3(1-c_\theta)+\hat{\omega}_2 s_\theta\\\hat{\omega}_1\hat{\omega}_2(1-c_\theta)+\hat{\omega}_3 s_\theta&c_\theta+\hat{\omega}_2^2(1-c_\theta)&\hat{\omega}_2\hat{\omega}_3(1-c_\theta)-\hat{\omega}_1 s_\theta\\\hat{\omega}_1\hat{\omega}_3(1-c_\theta)-\hat{\omega}_2 s_\theta&\hat{\omega}_2\hat{\omega}_3(1-c_\theta)+\hat{\omega}_1 s_\theta&c_\theta+\hat{\omega}_3^2(1-c_\theta)\end{bmatrix}$$

Joint angle limits for OpenSim were determined in part from RoM values listed by Platzer [20] and Kapandji [21]. The OpenSim model has a maximum hip flexion(+)/extension(-) of +140/-20, where the bipedal kinematics model is limited to +110, -15. Hip adduction(+) and abduction (-) is a function of hip flexion, and the difference in of Range of Motion (RoM) limits is demonstrated in Figure 3, below.

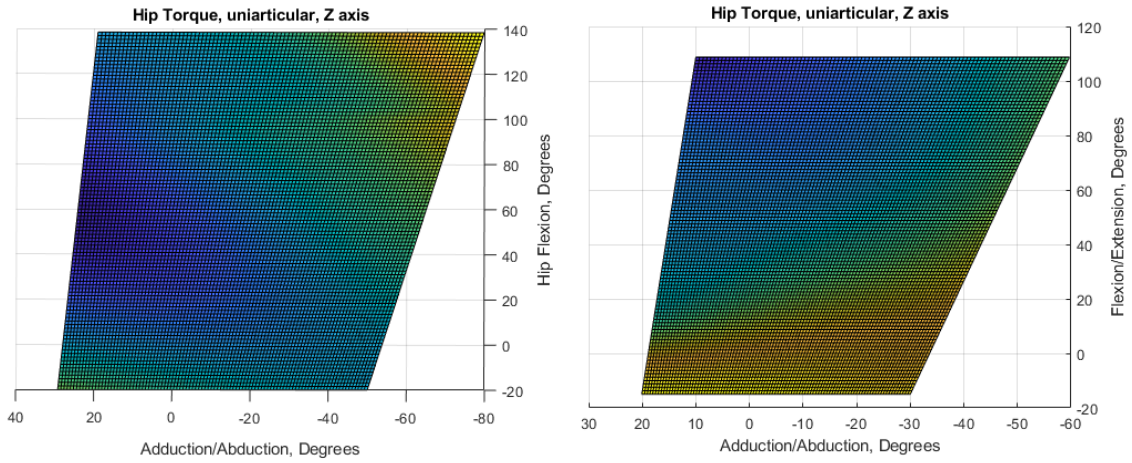


Fig. 3. RoM limits for hip flexion and adduction for OpenSim (left) and robot model (right).

2.2 Muscle Paths

Addition of muscles to the model is guided by GaitBody2392. OpenSim attachment points are kept the same, except the via points which are replaced by fixed points. Bipedal robot PAM attachment points are modified to accommodate the contraction limits of Festo PAM - and to try to accommodate 3D geometry realities (i.e. including avoiding interference with other actuators) - while also trying to maintain torques. Muscle path routings for GaitBody2392 are shown on the left hand side of Figure 4, below. PAM routings for the robot design are saved in OpenSim as *gait2392_robotbody.osim* (Fig. 4, right).

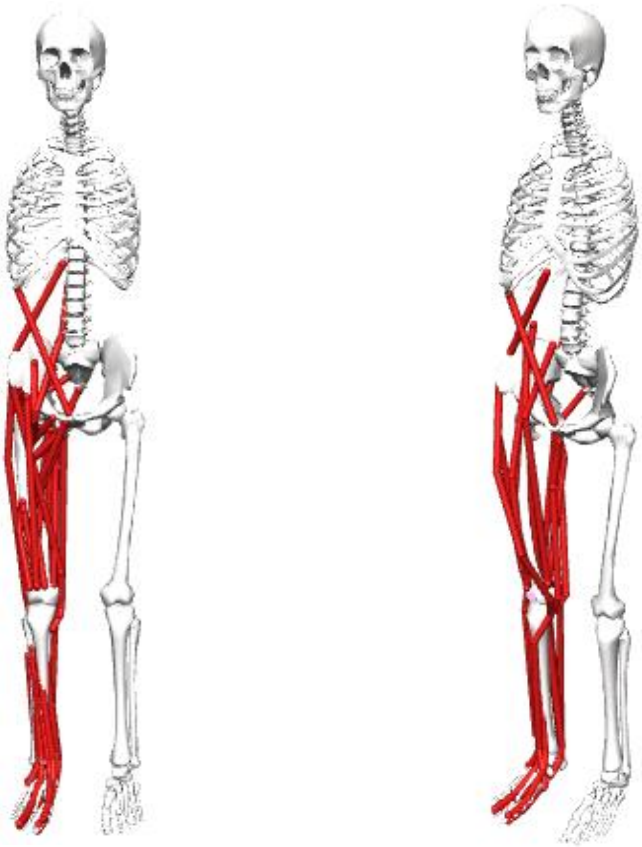


Fig. 4. Skeleton w/ OpenSim muscle paths (left) and PAM paths (right).

It is also important to note that that the Gait2392 model is not free of errors. At high degrees of hip flexion some muscle routings bunch up or pass through bone (Fig. 5).

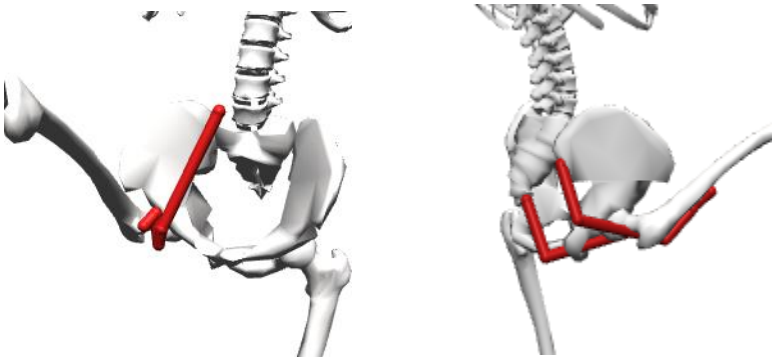


Fig. 5. Examples of muscle modelling error in GaitBody2392.

2.3 Muscle length calculations

All of skeletal muscles cross at least one joint. Therefore, the muscle origin, insertion, and wrapping points are described in different coordinate frames. All muscles cross at least one joint. Transformation matrices are used to transform coordinates into the same frame. The Euclidean norm is used to find the distance between two points. The sum of these norm segments are added up to find the musculotendon length. An example of adding Euclidean norms in Matlab for length calculations is

```
%Tensor Fasciae Latae
tfl_length(i,j) = norm(VecTrans(T_h,tfl_wr1)-tfl_o)+norm(tfl_wr2 - tfl_wr1)+norm(tfl_wr2-VecTrans(T_k,tfl_i)); %Total length, right
```

where T_h is the hip transformation matrix, tfl_o is the muscle origin in the Pelvis frame, and tfl_wr1 is a wrapping point described in the hip (femur) frame. In this example VecTrans(T_h, tfl_wr1) transforms tfl_wr1 into the pelvis frame from the hip frame, and returns its Cartesian coordinate in 3x1 vector form.

2.4 Muscle Moment Arm Calculations

The method to acquire muscle moment arms is described by Hoy [22] and further clarified by Sherman [23]. Moment arms are calculated in the body frame. This is shown in Figure 6, below.

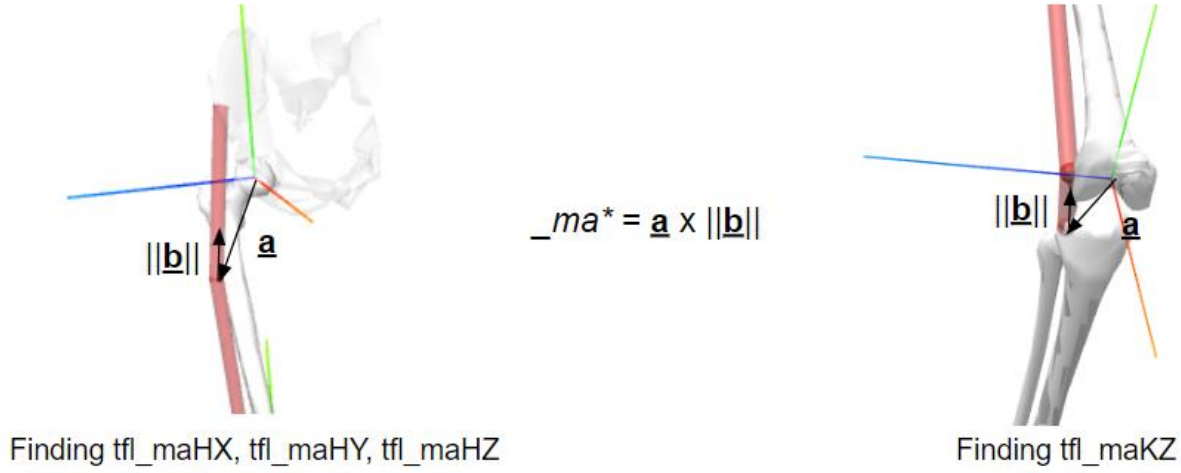


Fig. 6. Moment arm calculation.

This is calculated in Matlab as:

```
%Tensor Fasciae Latae
tfl_length(i,j) = norm(VecTrans(T_h,tfl_wrl)-tfl_o)+norm(tfl_wr2 - tfl_wrl)+norm(tfl_wr2-VecTrans(T_k,tfl_i)); %Total length, right
tfl_maHX(i,j) = CrossProd(tfl_wrl, (VecTrans(T_h\eye(4),tfl_o)-tfl_wrl)/norm(VecTrans(T_h\eye(4),tfl_o)-tfl_wrl),1); %Moment arm, Hip, x axis
tfl_maHY(i,j) = CrossProd(tfl_wrl, (VecTrans(T_h\eye(4),tfl_o)-tfl_wrl)/norm(VecTrans(T_h\eye(4),tfl_o)-tfl_wrl),2); %Moment arm, Hip, y axis
tfl_maHZ(i,j) = CrossProd(tfl_wrl, (VecTrans(T_h\eye(4),tfl_o)-tfl_wrl)/norm(VecTrans(T_h\eye(4),tfl_o)-tfl_wrl),3); %Moment arm, Hip, z axis
tfl_maKZ(i,j) = CrossProd(tfl_i, (VecTrans(T_k\eye(4),tfl_wr2)-tfl_i)/norm(VecTrans(T_k\eye(4),tfl_wr2)-tfl_i),3); %Moment arm, Knee, z axis
```

where the function CrossProd calculates the cross product of two vectors and returns the value for the axis of interest specified.

2.5 Muscle Size Calculations

Muscle size calculations are performed to determine the diameter and resting muscle length for the model attachment points and ranges of motion. Festo Group specifies a maximum contraction and force for their PAMs (Table 2 [24]).

Table 3: Festo Corp. parameters [24]

General technical data					
Size		5	10	20	40
Pneumatic connection		M3	G1/8	G1/4	G3/8
Design		Contracting diaphragm			
Mode of operation		Single-acting, pulling			
I.D.	[mm]	5	10	20	40
Nominal length	[mm]	30 ... 1000	40 ... 9000	60 ... 9000	120 ... 9000
Stroke	[mm]	0 ... 200	0 ... 2250	0 ... 2250	0 ... 2250
Max. additional load, freely suspended	[kg]	5	30	80	250
Max. permissible pretensioning[1)]		1% of nominal length	3% of nominal length	4% of nominal length	5% of nominal length
Max. permissible contraction		20% of nominal length	25% of nominal length		

The function *Size.m* requires the length matrix and maximum isometric force (mif). It returns inner diameter (I.D.) of the Festo muscle and the resting muscle length. It will also return the longest artificial musculo-tendon length, if requested. This value is needed when calculating force in Festo PAMs.

```
function [ID, size, longest]= Size(length, mif)
%function Size for sizing of Festo artificial muscles
%Inputs:
%Length is the musculotendon length, either a column vector or a matrix
%mif is Maximum Isometric Force
%Outputs:
%ID is inside diameter of the specified festo muscle
%size is the length of the festo muscle
%long is the longest artificial musculo-tendon length
```

The algorithm for determining the required Festo diameter is provided as follows:

```
F1 = 630;              %Maximum theoretical force for 10mm I.D. Festo muscle
F2 = 1500;             %Maximum theoretical force for 20mm I.D. Festo muscle
F3 = 6000;             %Maximum theoretical force for 40mm I.D. Festo muscle
Fc = 0.9;
x = 0.0125;            %Length of air fittings

if mif < Fc*F1
    ID = 10;
elseif mif >= Fc*F1 && mif < Fc*F2
    ID = 20;
elseif mif >= Fc*F2 && mif <= Fc*F3
    ID = 40;
else
    ID = 'Use additional muscles';
end
```

where Fc is a safety factor that prevents the design from getting too close to the maximum theoretical force.

The algorithm below shows how resting muscle length is calculated in the *Size.m* function. It initially aims for the Festo specified optimum contraction range of 9% between the longest and shortest musculotendon lengths. It then checks to see if this resting muscle length plus the length of two air fittings are shorter than the musculotendon length in its longest configuration. This is only a minimum requirement that does not take into account the amount of artificial tendon length that will be needed for certain actuator routing configurations. If this constraint is violated then the algorithm will incrementally adjust maximum allowable deformation, up to the Festo specified maximum strain of 25% (Table 3).

```
Find longest musculotendon length
Find shortest musculotendon length
delta = longest - shortest  (if delta is zero then display an error)
opt_max = 1.09;  %Maximum allowable length change for optimal performance (Festo)
max_def = 1.25;  %Maximum allowable length change (Festo)

size = delta/(1 - 1/opt_max);

if size+2*x > long
   for def = opt_max:0.001:max_def
      if size+2*x > long && def <= max_def
         size = delta/(1 - 1/def);
      elseif size+2*x <= long
         def
         break
      end
   end
end
```

2.6 Muscle Force Calculations

Force calculations for human muscles were determined from the work of Millard [25]; Thelen [26]; and Hoy [22]. Figure 7 shows the schematic for musculo-tendon equilibrium and normalized musculo-tendon force curves.

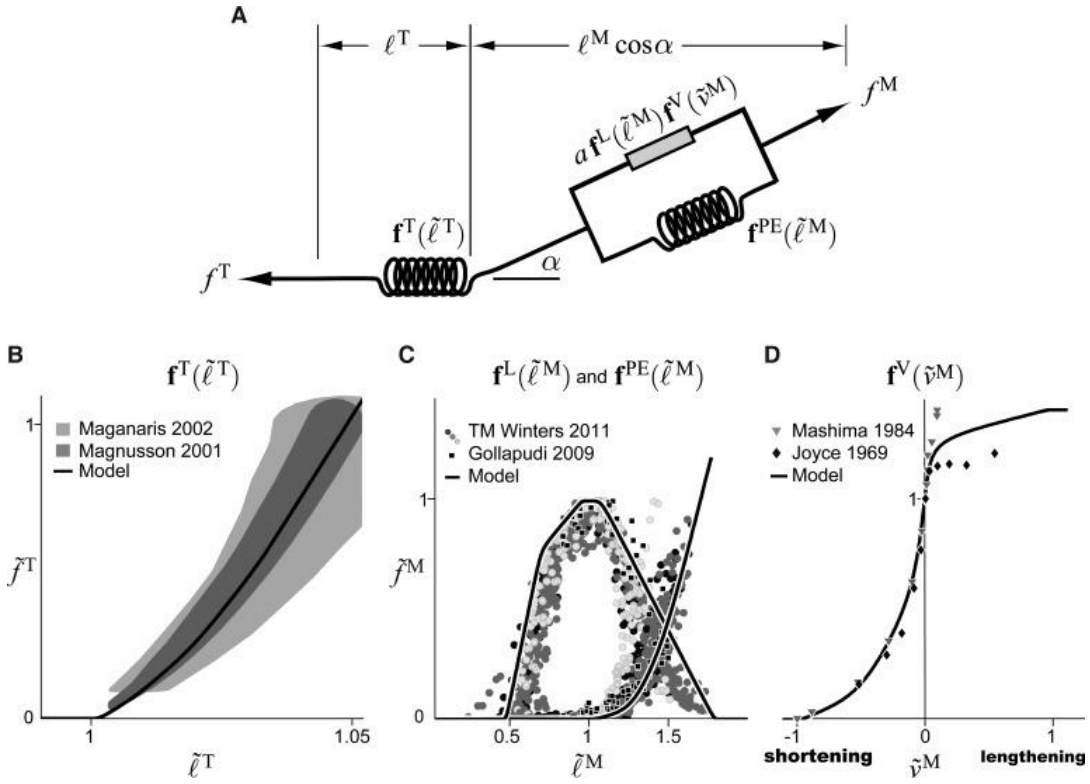


Fig. 7. Musculotendon equilibrium schematic (**A**), and data from experiments overlaid on a tendon-force–length curve (**B**), active- and passive-force–length curves (**C**), and force–velocity curve (**D**). Data points indicate experimental results. [Millard]

Total muscle force f^M is

$$f^M = f_o^M \left(a\mathbf{f}^L(\tilde{\ell}^M)\mathbf{f}^V(\tilde{v}^M) + \mathbf{f}^{PE}(\tilde{\ell}^M) \right) \quad (1)$$ [25]

where

f_o^M is the maximum isometric force at optimum fiber length.

$\tilde{\imath}^M$ is the normalized muscle length.

$f^L(\tilde{\imath}^M)$ is the normalized active muscle force.

$f^{PE}(\tilde{\imath}^M)$ is the normalized active muscle force.

a is activation time, a = 1 for static analysis.

$f^V(\tilde{v}^M)$ is normalized force lengthening or shortening, equal to 1 for static analysis.

Cosine of the pennation angle times the muscle force must balance the tendon force.

$$f_o^M \left(a\mathbf{f}^L(\tilde{\ell}^M)\mathbf{f}^V(\tilde{v}^M) + \mathbf{f}^{PE}(\tilde{\ell}^M) \right) \cos\alpha - f_o^M \mathbf{f}^T(\tilde{\ell}^T) = 0 \quad (2)\ [25]$$

where

$f^T(\tilde{\imath}^T)$ is the normalized tendon force for normalized tendon length $\tilde{\imath}^T$.

Eq. (3), below, shows how cos(α) is calculated when α_o (pennation angle at optimum muscle length) is known. Eq. (4) is the equation for normalized tendon force.

$$\bar{F}^T = \bar{F}^M(\bar{l}^M)\cos\alpha = \bar{F}^M(\bar{l}^M)\sqrt{1-\left(\frac{\sin\alpha_0}{\bar{l}^M}\right)^2} \quad (3)\ [22]$$

$$\tilde{F}^T = \left(\frac{37.5}{\bar{l}_s^T}\right)\left(\bar{l}^{MT} - \bar{l}^M\sqrt{1-\left(\frac{\sin\alpha_0}{\bar{l}^M}\right)^2} - \bar{l}_s^T\right). \quad (4)\ [22]$$

Eqs. (5) and (6) are for the normalized passive force and normalized active force, respectively.

$$\bar{F}^{PE} = \frac{e^{k^{PE}(\bar{L}^M-1)/\varepsilon_0^M} - 1}{e^{k^{PE}} - 1} \quad (5)\ [26]$$

$$f_l = e^{-(\bar{L}^M-1)^2/\gamma} \quad (6)\ [26]$$

where

$k^{PE} = 4$ is the passive exponential shape factor (OpenSim *Kshape Pasive*).

$\varepsilon_o^M = 0.6$ is the passive muscle strain at maximum isometric force (OpenSim *FmaxMuscleStrain*).

$\gamma = 0.5$ is the active shape factor (OpenSim *KshapeActive*).

Festo PAM force vs relative strain tables were provided by Dr. Hunt. Equation is given by

$$P = a_0 + a_1 * tan\left(a_2\left(\frac{k}{a_4 * F + k_{max}} + a_3\right)\right) + a_5 * F + a_6 * S \quad (7)\ [27]$$

Table 4: List of parameters for eq. (7) [27]

a_0 kPa	a_1 kPa	a_2	a_3	a_4 $1/_{mN}$	a_5 ($kPa/_{mN}$)	a_6 kPa
254.3	192.0	2.0265	-0.461	-0.331	1.230	15.6

For the Festo force function *festo.m* the maximum strain kmax is a function of the resting muscle length (rest). It varies from 14.91% to 17.50%. The current strain k is

```
k = (rest-(Lmt-tendon))/rest; %current strain
```
(8)

where Lmt is the current musculo-tendon length and tendon is the length of artificial tendon. Relative strain then is just

rel = k/kmax (9)

2.7 Muscle Torque Calculations

Torque is calculated once muscle force and moment arm are known (Eq. 10)

Torque = force*moment arm (10)

2.8 Matlab algorithm

Developed *OpenSim_kinematics.m* in Matlab to mimic the calculations in OpenSim. Via points not used because they require a *for* loop to add an attachment point for certain joint angles, requiring separate length and moment calculations. These calculations were checked against OpenSim results as a proof-of-concept. The algorithm for *OpenSim_kinematics.m* (herein called OpenSim) is

Call attachment points
Set RoM limits, joint movements

2 for loops
➔ T matrix
➔ Length
➔ Moment arm
➔ Force
➔ Torque calculations
end

Surface figures
Size calculations

The program *bipedal_kinematics.m* was created to calculate length, moment arm, force, and torque for the robot using PAMs. It is a similar algorithm to *OpenSim_kinematics*, but incorporates the Festo muscle force lookup table [27]. Force calculations in *bipedal_kinematics* require resting muscle length, which requires that muscle's length matrix to be computed first. The algorithm for *bipedal_kinematics* is

Call attachment points
Set RoM limits, joint movements

2 for loops
➔ T matrix
➔ Length
➔ Moment arm
end

Size calculations

2 for loops
➔ Force
➔ Torque calculations
end

Surface figures

3. Results

Table 5. Resting Muscle length and diameter for robot model.

	Resting muscle length, m				
Festo diameter (mm)	10	20	40	Motion	Group
Glut Max			0.644	Hip adduction and flexion	uni_hip
Semimembranosus			1.026	Hip Flex and knee flex	bi_hip/knee
Bicep Femoris, LH		0.8463		Hip Flex and knee flex	bi_hip/knee
Bicep Femoris, SH		0.3365		Hip Flex and knee flex	bi_hip/knee
Sartorius	1.5008			Hip Flex and knee flex	bi_hip/knee
Adductor Magnus			0.6105	Hip adduction and flexion	uni_hip
TFL	0.8427			Hip Flex and knee flex	bi_hip/knee
Gracilis	0.7738			Hip Flex and knee flex	bi_hip/knee
Iliacus		0.5244		Hip adduction and flexion	uni_hip
Psoas		0.5302		Hip adduction and flexion	uni_hip
Rectus Femoris		0.693		Hip Flex and knee flex	bi_hip/knee
Vastus Int			0.4075	Hip Flex and knee flex	bi_hip/knee
Medial Gastro			0.4643	Knee and ankle flex	calves
Lateral Gastro		0.4809		Knee and ankle flex	calves
Soleus			0.3576	Knee and ankle flex	calves
Tibialis Posterior			0.3826	Ankle flex and eversion	foot
Flex. Digit. Longus	0.2874			Ankle and toe flex	toe
Flex. Hallucis Long.	0.3682			Ankle and toe flex	toe
Tibialis Anterior		0.4222		Ankle flex and eversion	foot
Peroneus Brevis	0.3776			Ankle flex and eversion	foot
Peroneus Longus		0.3931		Ankle flex and eversion	foot
Peroneus Tertius	0.3993			Ankle flex and eversion	foot
Ext. Digit. Long.	0.522			Ankle and toe flex	toe
Ext. Hallucis Long.	0.5327			Ankle and toe flex	toe
Erector Spinae			0.2703	Back Flex and bending	ercspn
Internal Oblique		0.3527		Back rotation and bending	ab
External Oblique		0.2697		Back rotation and bending	ab
Total (both sides)	11.209	9.698	8.3256		

Table 5 shows the Festo diameter and resting PAM lengths calculated for the robot model's 27 actuators. These values were calculated for the DoF movement listed under the *Motion* column. Some of these muscles are long and require two PAMs to be placed in series. The resting length is displayed as total length per muscle per side. The *Total* row accounts for both right and left side muscles.

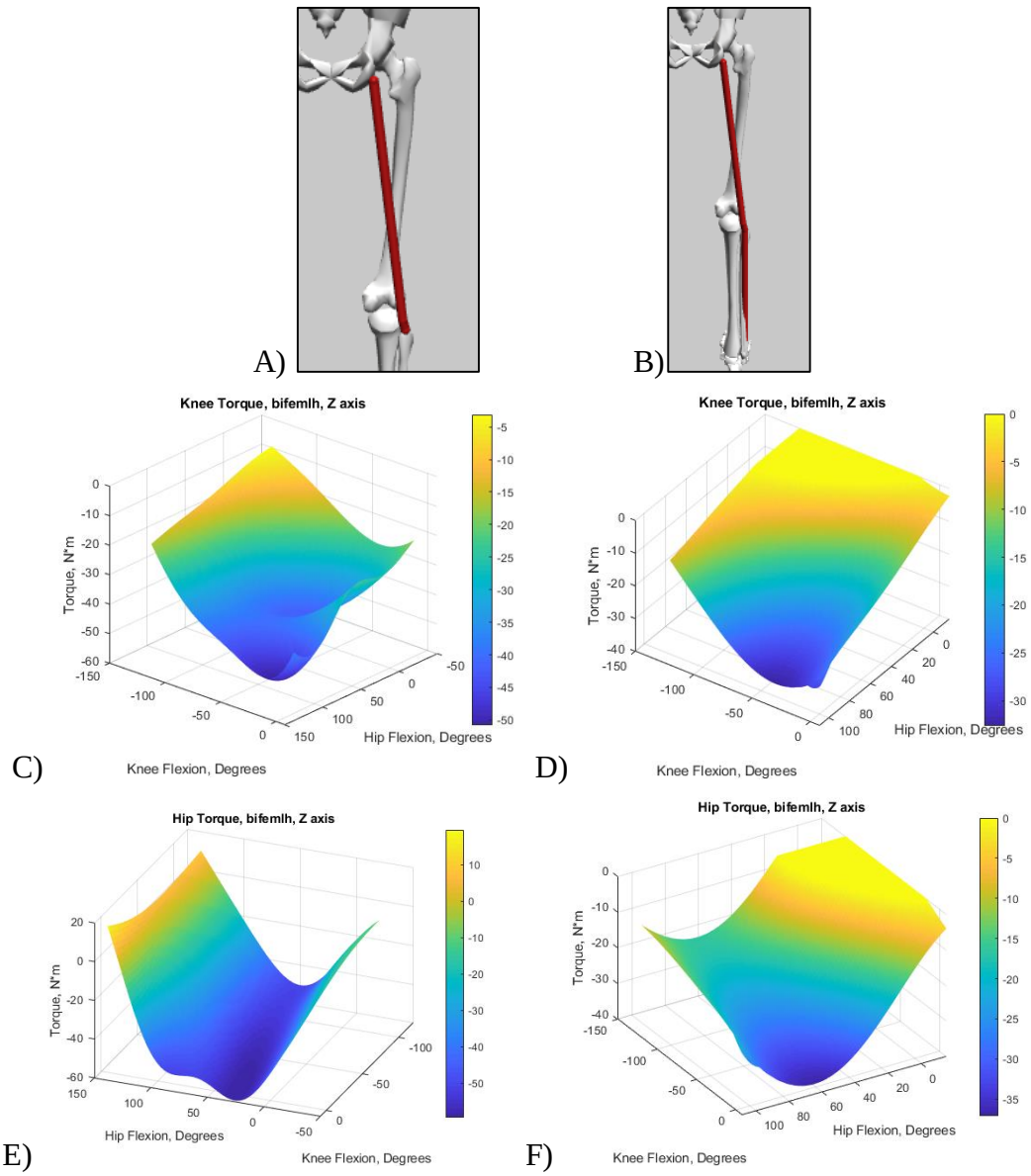


Fig. 8. Biceps Femoris Long Head muscle path (a & b) and torque values (c - f) for OpenSim (a, c, e) and bipedal_kinematics (b, d, f), for knee (c, d) and hip Z-axis (e, f).

Fig. 8. shows the difference in the musculotendon paths used in OpenSim and the robot models for the long head of the biceps femoris muscle actuator. The action of this biarticular muscle is, in part, to perform hip extension and knee flexion. Therefore, torque is examined for the knee Z-axis and the hip Z-axis through over the range of possible hip and knee positions.

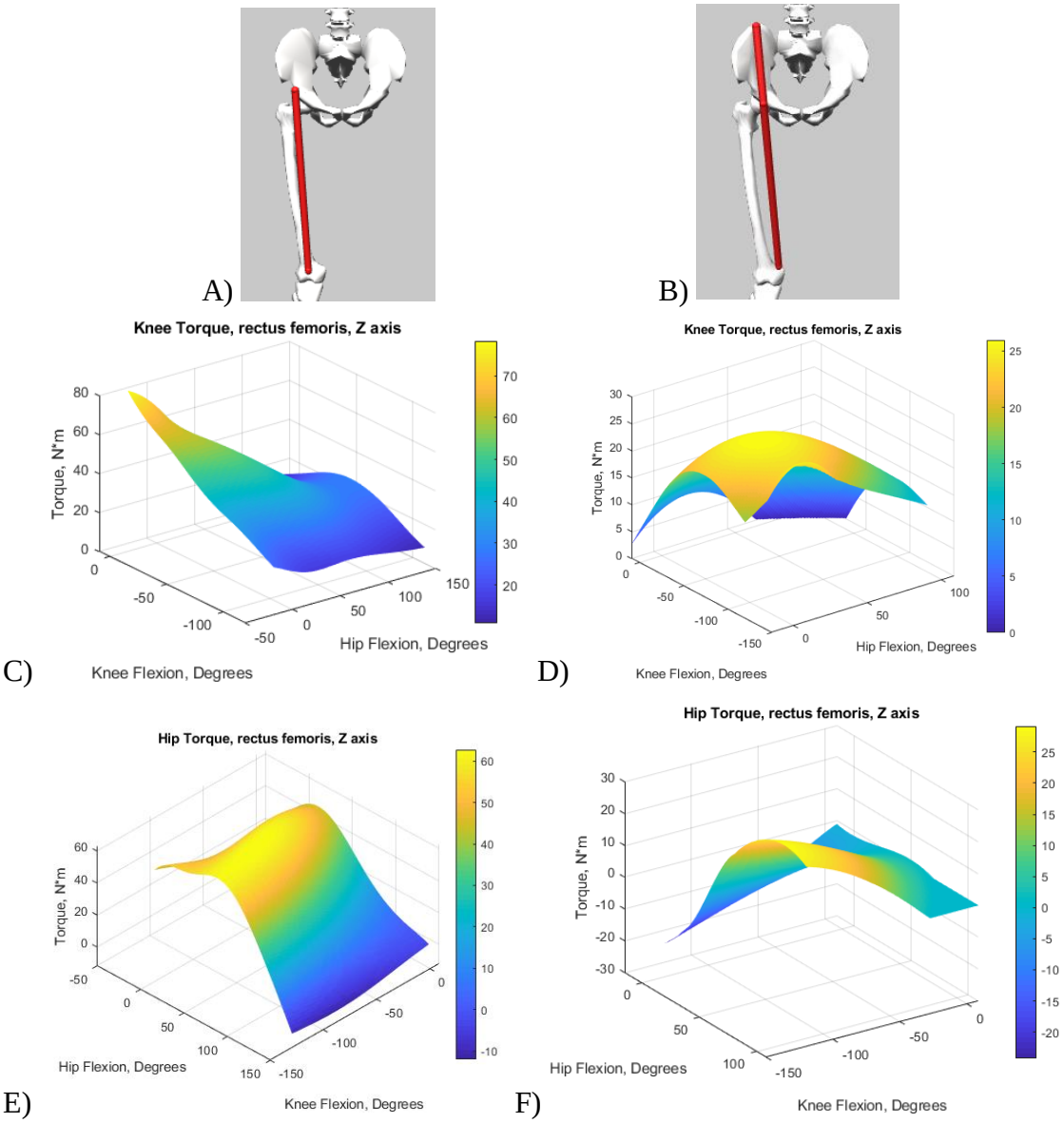


Fig. 9. Rectus femoris muscle paths (a & b) and torque values (c - f) for OpenSim (a, c, e) and bipedal_kinematics (b, d, f), for knee (c, d) and hip Z axis (e, f).

Figure 9 compares the different rectus femoris muscle paths. The rectus femoris is the biarticular muscle of the quadriceps group. Its action is hip flexion and knee extension. Therefore, torque is compared for hip and knee flexion.

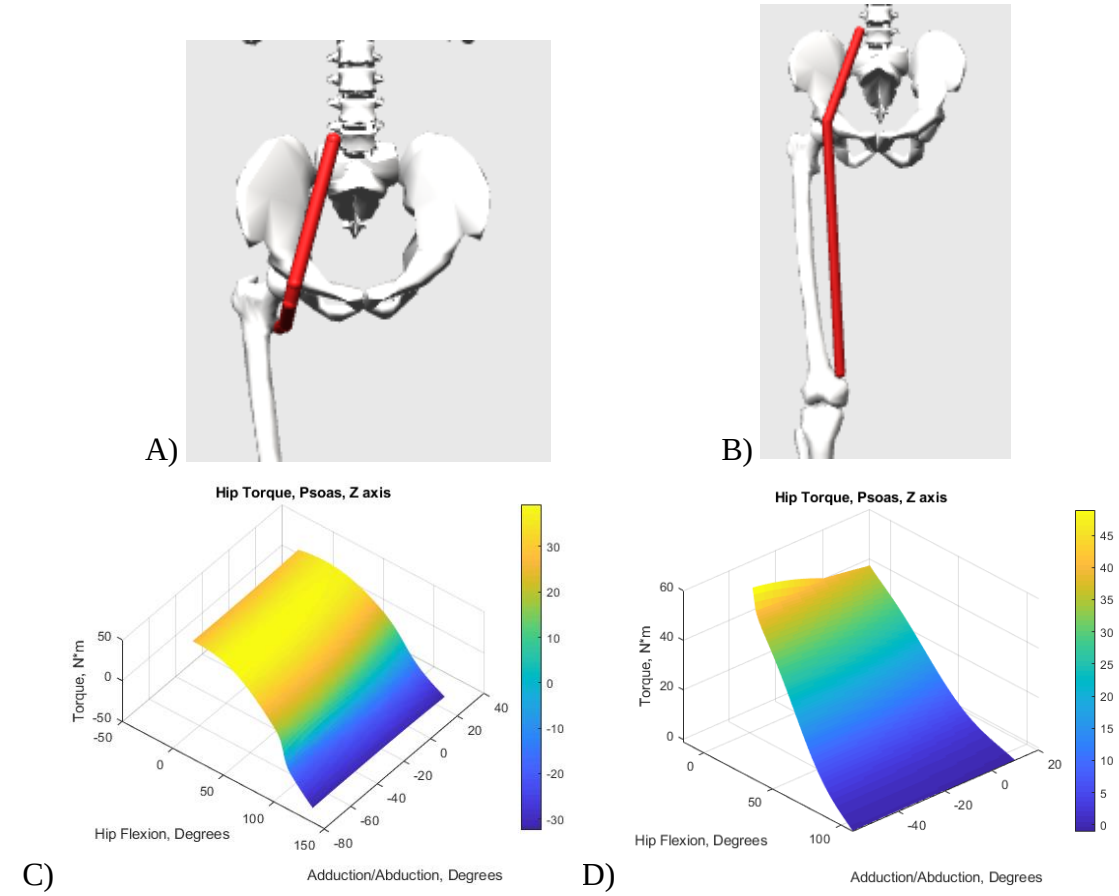


Fig. 10. Psoas muscle path (a & b) and torque values about the hip Z-axis (c & d) for OpenSim (a & c) and bipedal_kinematics (b & d).

Figure 10 shows the psoas muscle, modeled as a uniarticular hip flexor in OpenSim. In bipepal_kinematics it is a biarticular muscle that attaches to the lumbar vertebrae. Torque over the hip Z-axis and muscle paths are compared. The longest and shortest muscle configurations can be approximated better by looking at adduction/abduction and flexion/extension than if one of these DoFs were replaced by external/internal rotation.

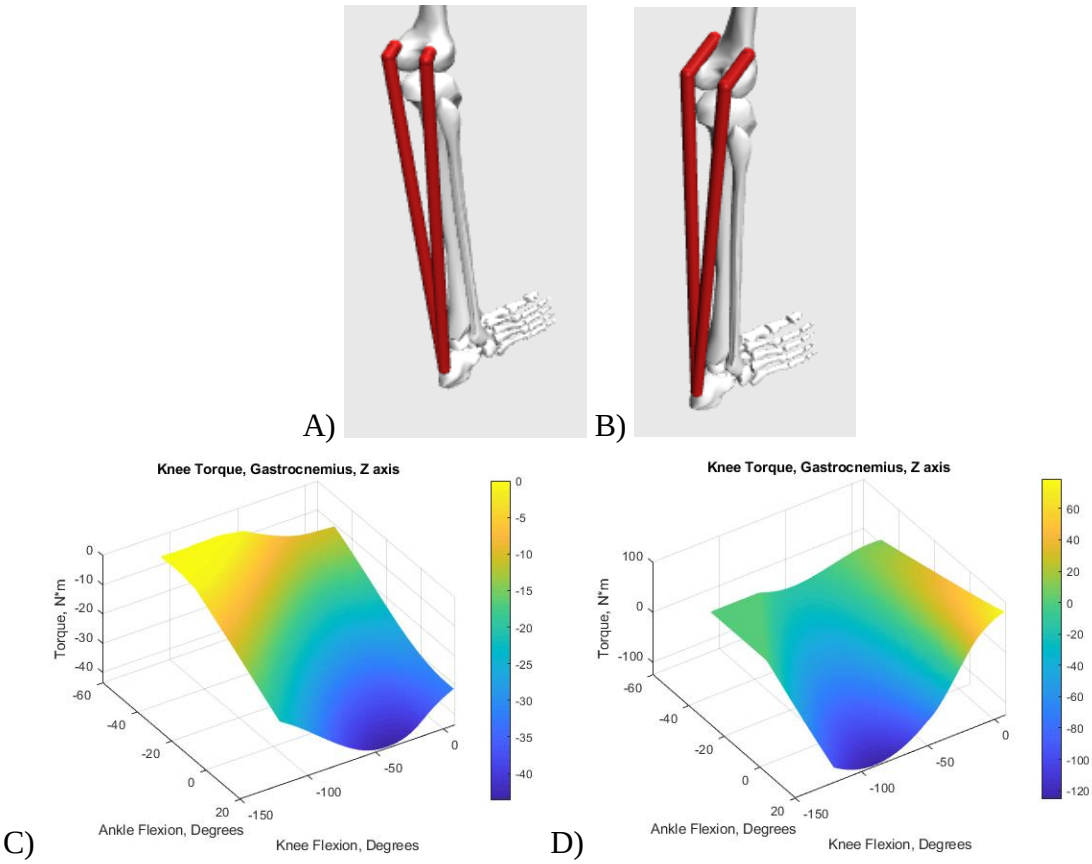


Fig. 11. Gastrocnemius muscle paths (a & b) and knee torque values (c & d) for OpenSim (a & c) and bipedal_kinematics (b & d).

The gastrocnemius is a two-headed tri-articular calf muscle modeled as two separate linear actuators in OpenSim and bipedal_kinematics (Fig. 11). Torque is compared over the knee Z-axis for these actuators for different knee and ankle orientations. It is assumed that its torque about the subtalar X-axis does not need to be investigated. Another assumption is that investigating ankle and knee flexion will produce a good idea of the longest and shortest lengths the model may encounter.

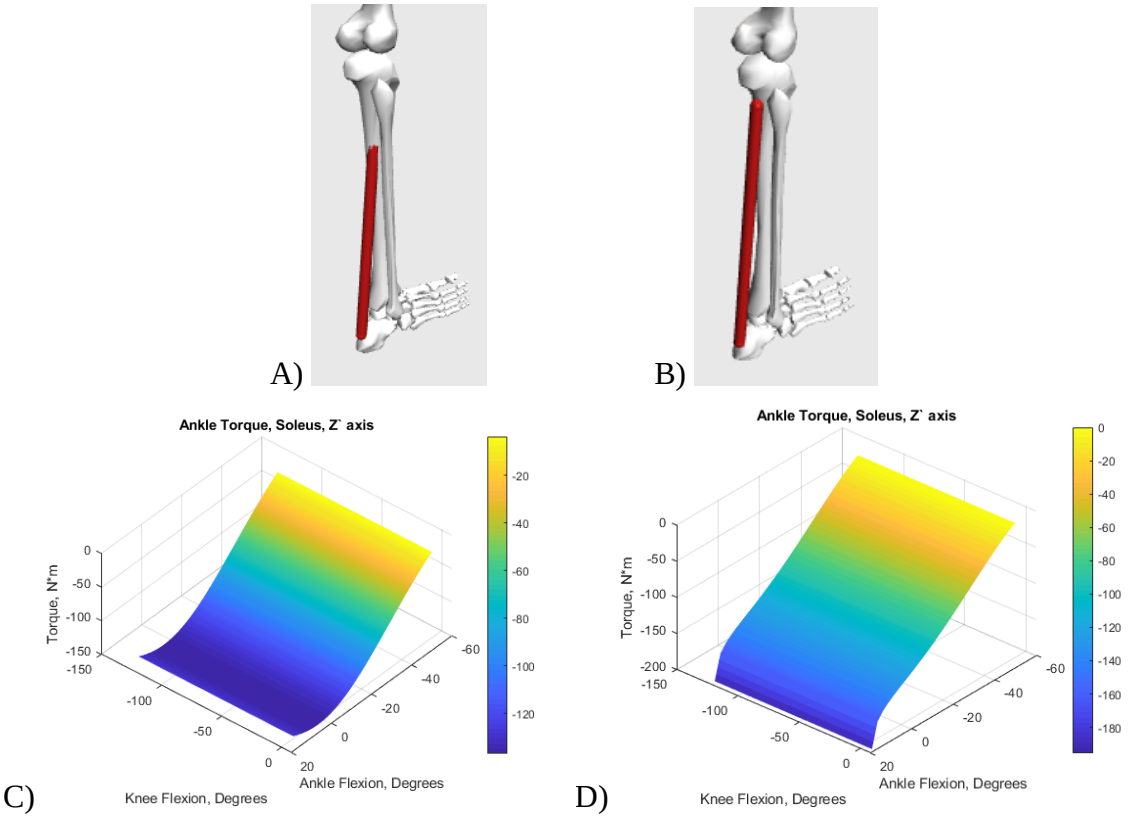
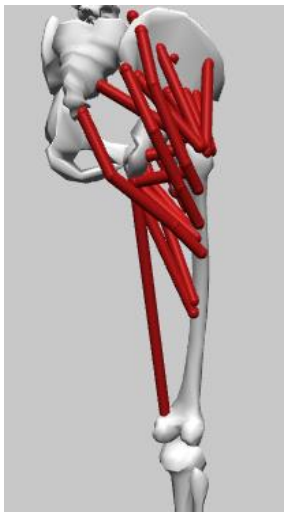


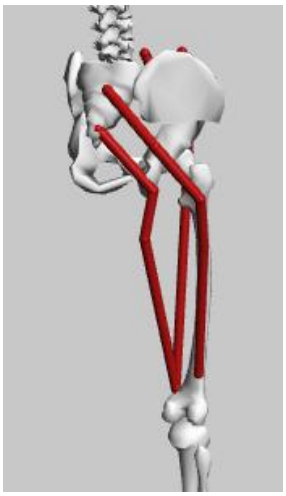
Fig. 12. Soleus muscle path (a & b) and Ankle torque values (c & d) for OpenSim (a & c) and bipedal_kinematics (b & d).

The soleus (Fig. 12) is a biarticular muscle that assists in ankle plantarflexion and dorsiflexion. The soleus origin was moved higher up on the tibia in the bipedal_kinematics model. The insertion point on the calcaneus was kept the same.

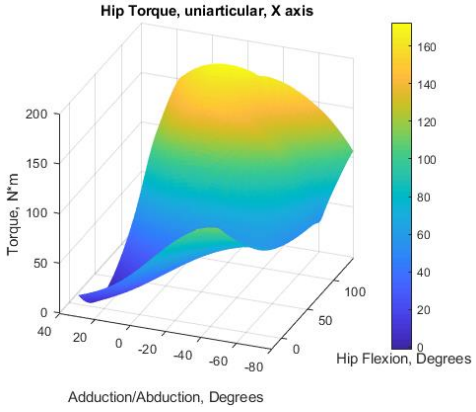
A)



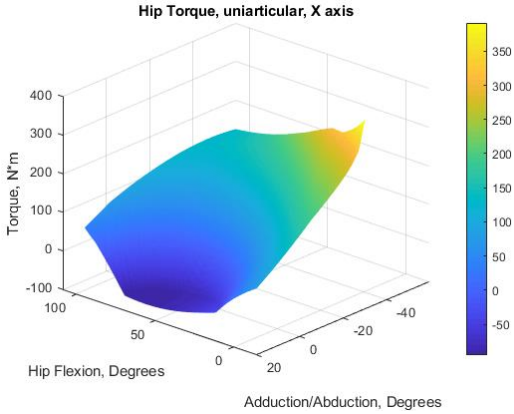
B)



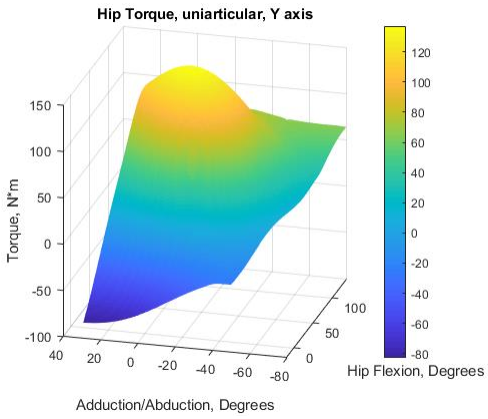
C)


Hip Torque, uniarticular, X axis
Torque, N*m
Hip Flexion, Degrees
Adduction/Abduction, Degrees

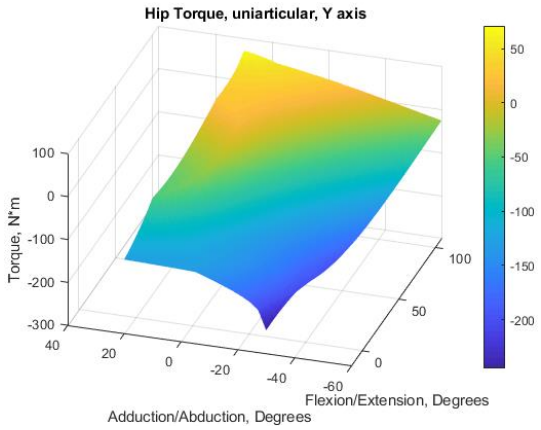
D)


Hip Torque, uniarticular, X axis
Torque, N*m
Hip Flexion, Degrees
Adduction/Abduction, Degrees

E)


Hip Torque, uniarticular, Y axis
Torque, N*m
Hip Flexion, Degrees
Adduction/Abduction, Degrees

F)


Hip Torque, uniarticular, Y axis
Torque, N*m
Flexion/Extension, Degrees
Adduction/Abduction, Degrees

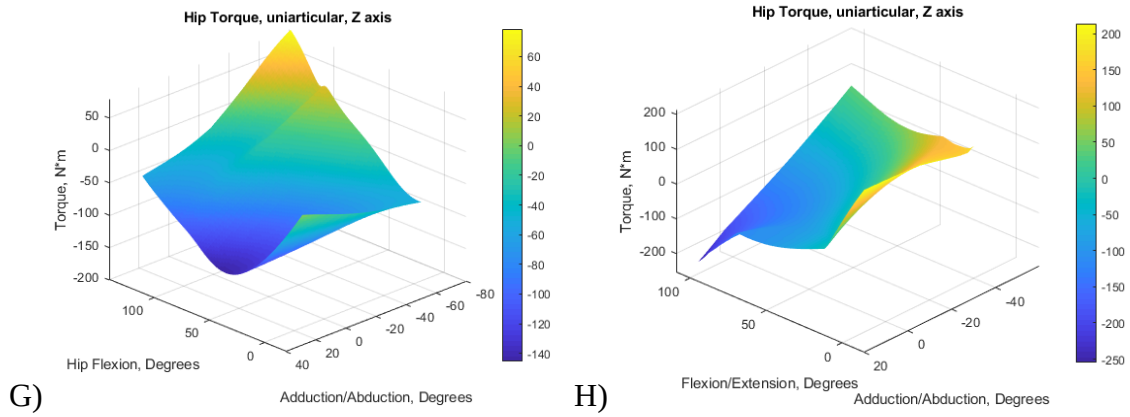
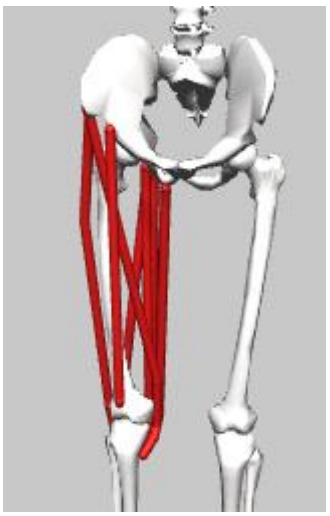
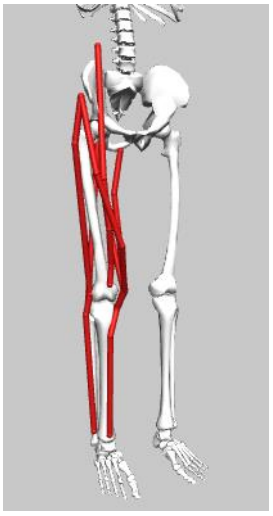
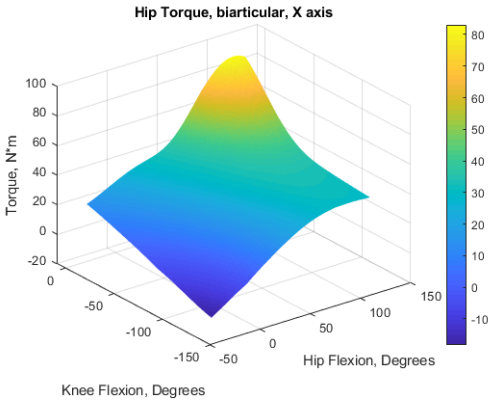


Fig. 13. Uniarticular hip muscle paths (a & b) and torque values (c - h) for OpenSim (a, c, e, g) and bipedal_kinematics (b, d, f, h), for hip X-axis (c, d), Y-axis (e, f), and Z-axis (g, h).

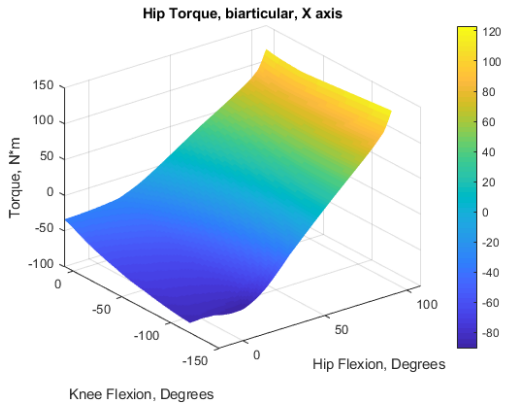
Fig. 13 groups and compares the uniarticular muscles of the hip joint (including the psoas). Since many of the gluteus, adductor, and external hip rotators from the OpenSim model were removed in bipedal_kinematics model, it is important to examine the group as a whole and see the if the proposed changes in routings and muscle consolidation can provide similar torque. Hip rotation will not affect length as much as the other two hip DoFs and therefore was not investigated.

A) 

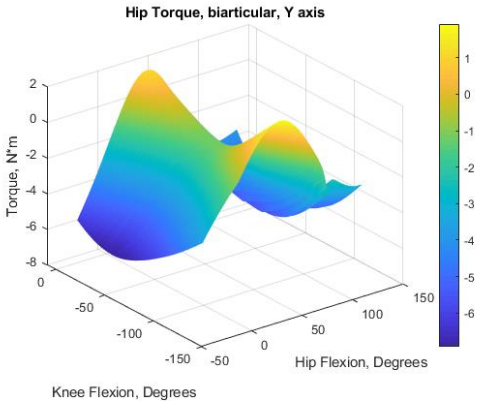
B) 


Hip Torque, biarticular, X axis
Torque, N*m
Hip Flexion, Degrees
Knee Flexion, Degrees

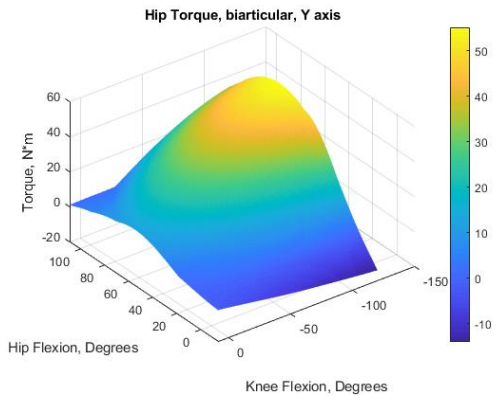
C)


Hip Torque, biarticular, X axis
Torque, N*m
Hip Flexion, Degrees
Knee Flexion, Degrees

D)


Hip Torque, biarticular, Y axis
Torque, N*m
Hip Flexion, Degrees
Knee Flexion, Degrees

E)


Hip Torque, biarticular, Y axis
Torque, N*m
Hip Flexion, Degrees
Knee Flexion, Degrees

F)

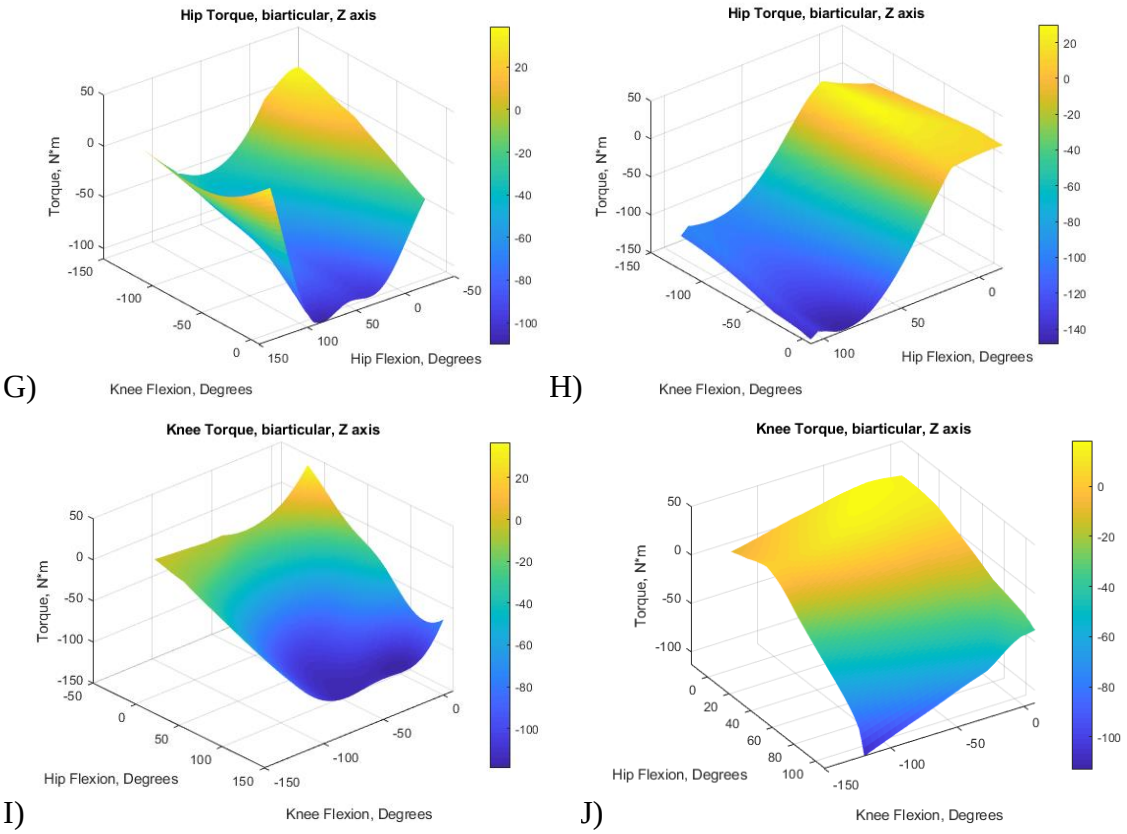


Fig. 14. Biarticular hip/knee muscle paths (a & b) and torque values (c - h) for OpenSim (a, c, e, g) and bipedal_kinematics (b, d, f, h), for hip X-axis (c, d), Y-axis (e, f), Z-axis (g, h), and knee Z-axis (i, j).

Fig. 14 shows the biarticular muscles that cross the hip and knee joint. The muscle paths and torque values are show for hip and knee flexion.

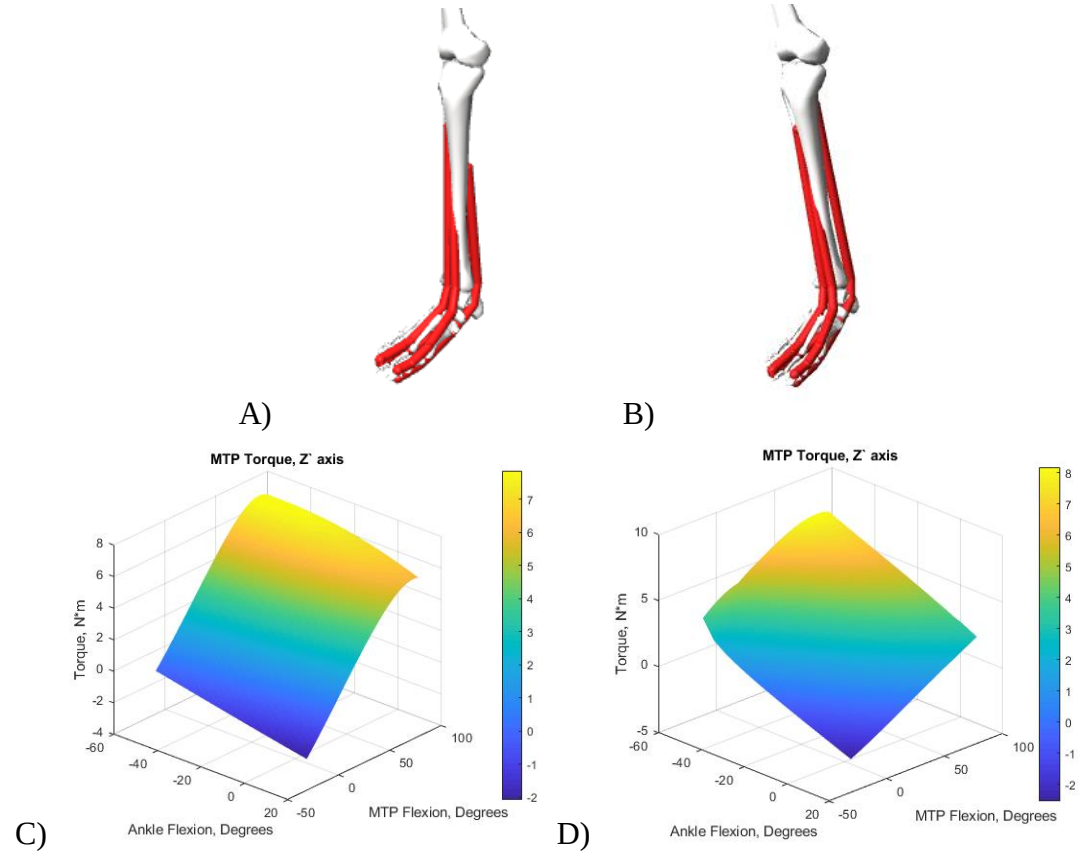


Fig. 15. Toe muscle paths (a & b) and torque values (c & d) for OpenSim (a & c) and bipedal_kinematics (b & d).

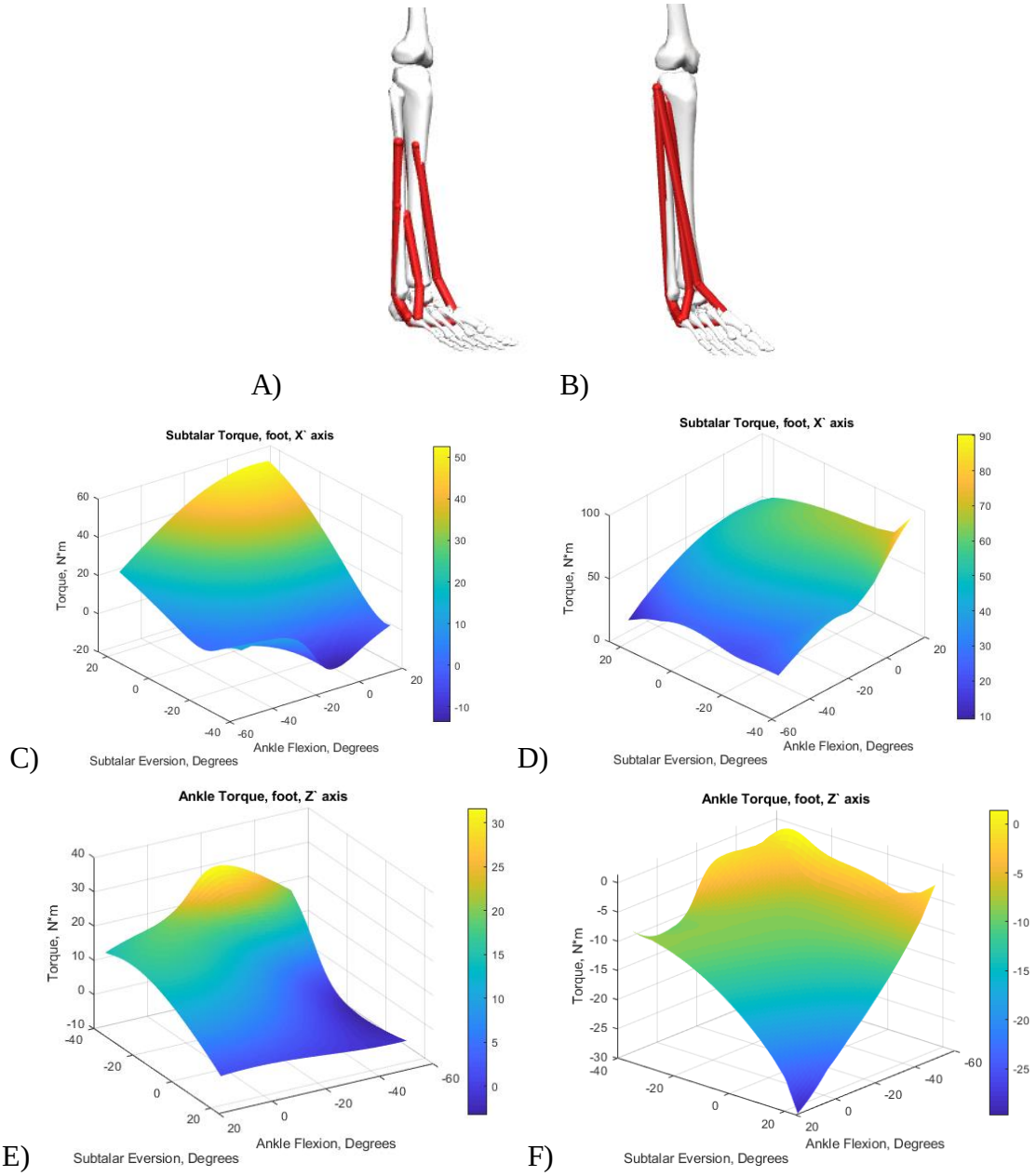


Fig. 16. Foot muscle groupings (a & b) and torque values (c - f) for OpenSim (a, c, e) and bipedal_kinematics (b, d, f), for subtalar X-axis (c, d) and ankle Z-axis (e,f).

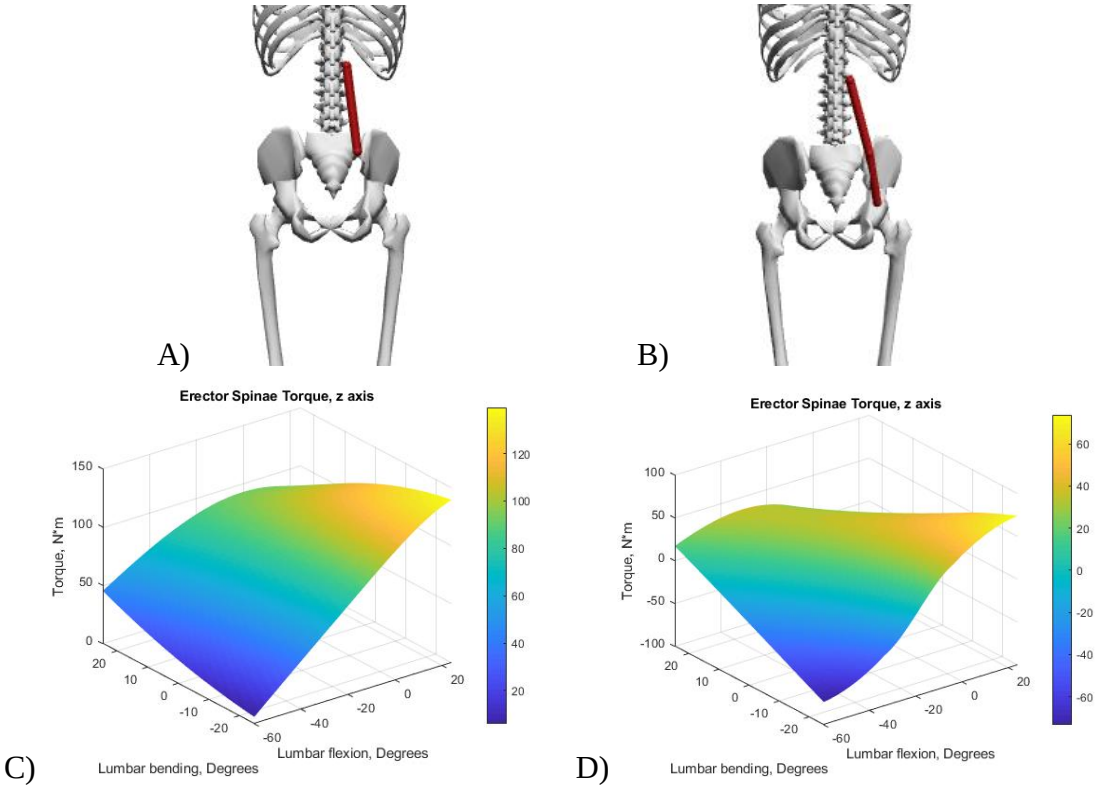


Fig. 17. Erector Spinae muscle routing (a & b) and torque values (c & d) for OpenSim (a & c) and bipedal_kinematics (b & d) for the back Z-axis.

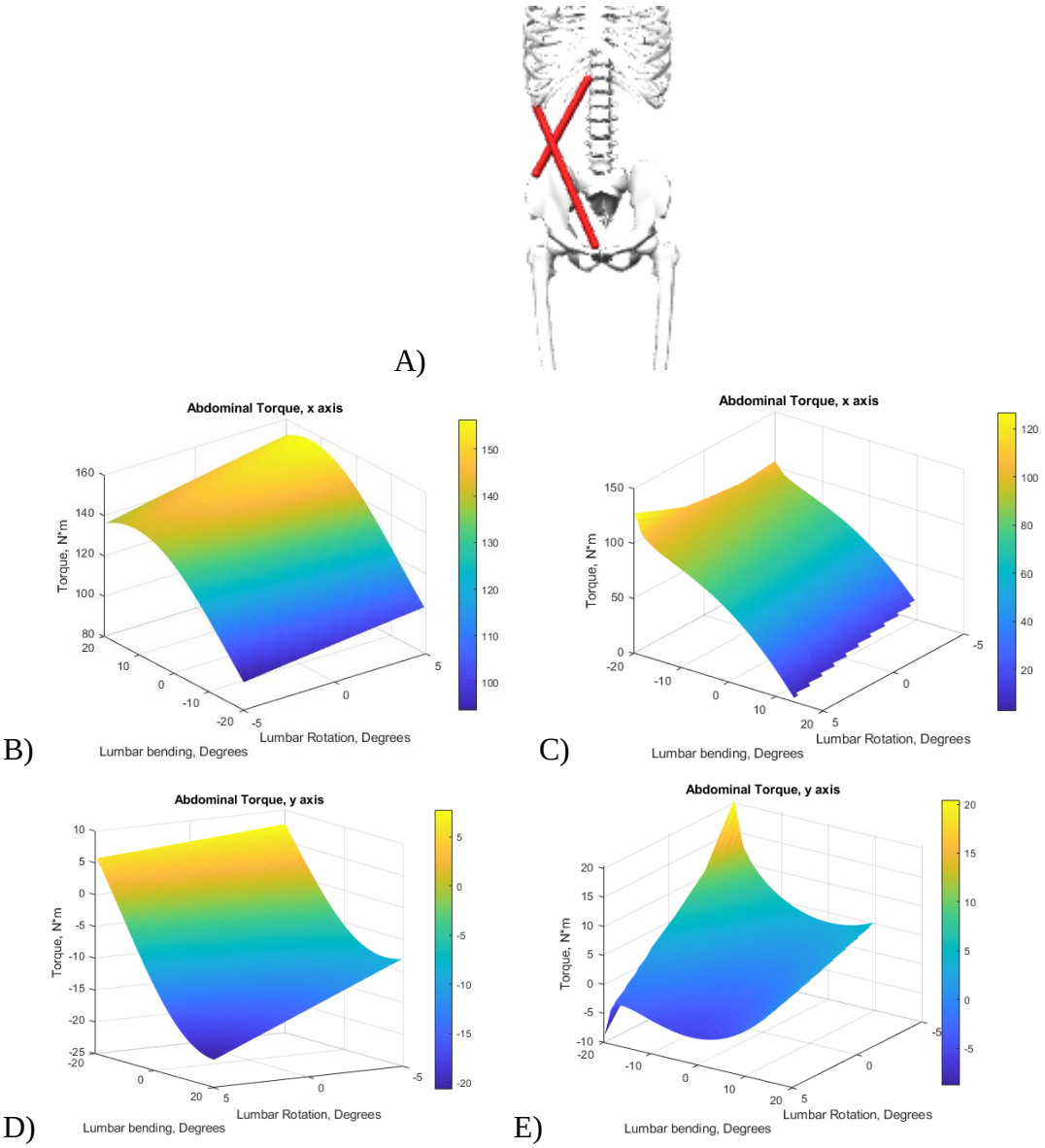


Fig. 18. Abdominal muscle groupings (a) and torque values (b - e) for OpenSim (b,d) and bipedal_kinematics (c, e), over the back X-axis (b, c) and Y-axis (d, e).

4. Conclusion

The replication of OpenSim calculations in Matlab was successful. Small variations were observed due to a simplification of force calculations and replacing via points with fixed attachment points. The OpenSim Matlab model *OpenSim_kinematics* takes more time to compute values than OpenSim, but has the benefit of being able to do

surface plots of calculated values over 2 degrees of freedom. This can be expanded to analyzing more than 2 degrees of freedom at once. The bipedal kinematics Matlab model *bipedal_kinematics* is a lot faster than *OpenSim_kinematics* because the force function *festo* uses a lookup table. The speed of the OpenSim_kinematics force calculation *forz* can be increased by incorporating assumptions of muscle deactivation when the muscle is unrealistically short ($l^M < l^M_{min}$).

It is a challenge to achieve realistic joint RoMs and stay within the Festo specified limit of 25% maximum contraction. This difficulty is increased when taking into account that Hunt [27] reports a maximum strain of 17.5% for long resting lengths of the 10 mm Festo PAM. The force function *festo* returns a value of zero when relative strain rel is outside the range of zero to one. This is be interpreted as the PAM either buckling or breaking. Figure 19 gives some examples of potential hose kinking (flat areas are called dead zones).

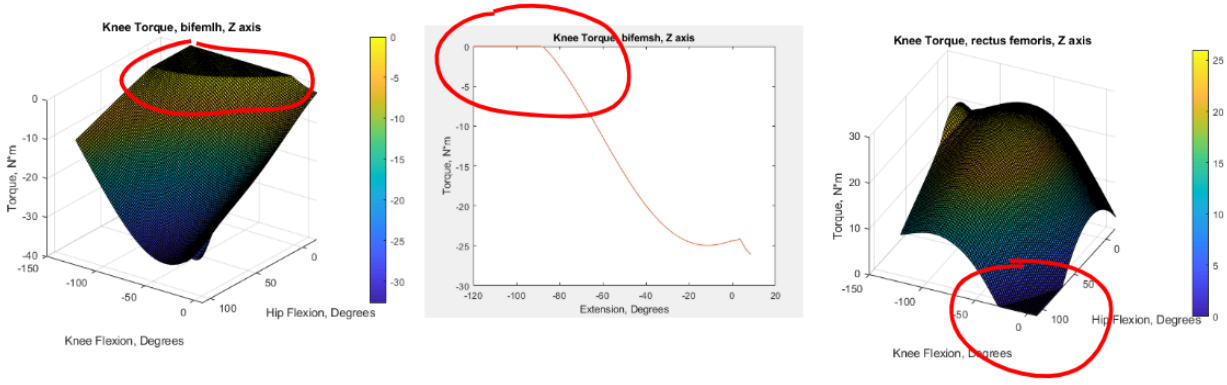


Fig. 19. Joint angles where Size function breaks down for certain muscles

Festo hose diameters of 20 mm and 40 mm should be tested to see if they do indeed have the same maximum strain as the 10 mm Festo muscle. The assumption that their force output will upscale based on maximum theoretical force should also be tested.

These tests will increase the accuracy of the robot model. The Size function can be modified to incorporate the maximum strain reported by Hunt.

Many individual PAMs and muscle-actuator groups have torque values that have surface plots with significantly different shapes and torque ranges than the OpenSim model. The biceps femoris change in length requirement was achieved by changing the muscle origin, insertion, and wrapping point, but there are dead zones in the graph. Since this means that the maximum strain is between 17.5 – 25%, it can be inferred that moving the origin point up the sacrum and better modeling of the 3D geometry in Matlab (with wrap points) might eliminate these dead zones

Rectus femoris torque values do not match up very well. This comparison also note that attachment modifications lead to a longer musculo-tendon length in the hip flexed position (Fig. 20). This was done to reduce the delta of length.

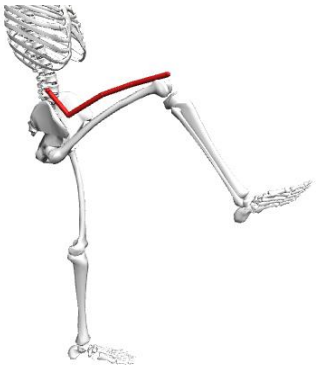


Fig. 20. Rectus Femoris PAM routing during knee flexion

The psoas model for bipedal kinematics does not match up very well with the model in OpenSim (Fig. 10). Part of this can be explained by the modeling error for this muscle in OpenSim (Fig. 5). Addition of via points in OpenSim_kinematics might

produce a better model, and the moving the psoas insertion in bipedal_kinematics further up and towards the rear of the femur should be investigated.

The gastrocnemius and soleus in the robot model can resist much more torque than the one in the OpenSim model (Figs. 11 and 12, respectively). This can be changed by reducing one of the actuator diameters or increasing the actuator resting length (if possible). The gastrocnemius also has positive torque value for certain ankle and knee position. This can be eliminated by moving the attachment points out and away from the back of the knee. Solid modelling the bone geometry and PAMs will aid in this effort.

Looking at the groups of uniarticular (Fig. 13) and biarticular (Fig. 14) hip muscles it can be seen that there is not yet a close fidelity between the OpenSim model and bipedal_kinematics. It deserves reiterating that the short length muscles in OpenSim can deactivate when their length gets too short, and that there are certain errors when hip flexion or knee flexion become too great. Solid-modeling PAM routings can be used to get more accurate muscle paths. Additional adductor/abductor hip muscles can be added, and attempts can be made to further reduce RoM limits.

The MTP joint comparison the surface plots have similar shapes and have a maximum torque of about 8 N-m at high MTP flexion and ankle dorsiflexion, and -2 N-m at low MTP flexion and ankle plantarflexion (Fig. 15). Further investigation into these muscles would also compare the torque they can resist at the ankle and subtalar joints, and how this balances with other muscles that cross those joints. Future design iterations would take into account 3D geometry and wrapping of these muscles, and make sure

there is not a section of PAM that would wrap under the foot or bunch up near the ankle joint.

The PAMs that have their origin on the tibia and terminate on the calcanus (subtalar frame) do not match well with the OpenSim model (Fig. 16). This may be that certain muscles are too powerful and need to be either consolidated or have their resting lengths increased. Interference between actuators is not taken into account for these actuator paths.

The change in the muscle path for the erector spinae muscles (Fig. 17) was done so that the resting length would stay within the contraction limit. Looking at the differences in the muscle path it may seem like a minor routing change, but the PAM moment arm is changed and now produces a negative torque during lumbar flexion and negative lumbar bending. This results in a very different torque curve. The fix can be achieved by moving the wrapping point and adding one or two more.

Ab muscles torque plots have similar shape and values between bipedal_kinematics and OpenSim (Fig. 18). The values near the RoM limits appear to not match up. The action of the internal and external obliques is lumbar bending and rotation, so those were the DoFs investigated for them. It should be noted, however, that lumbar flexion/extension will have more of an effect on the maximum and minimum lengths, and that these Festo muscle calculations will break down when including this DoF (and excluding lumbar rotation).

Range of motion limitations placed on the robot model helped the Size function to specify acceptable resting lengths for the PAMs. Hip joint angles for flexion/extension

were originally set to match the OpenSim model but after experimentation the limits were set at +110/-15 (respectively). Maximum adduction at max hip flexion is +10 and at max hip extension it is +20. Maximum abduction at max hip flexion is -60 and at max hip extension it is -10. Further reducing of joint angle RoM was not as effective in achieving desired lengths when compared to moving the origin, insertion, and wrapping points for muscle actuators.

The bipedal robot design has 27 pneumatic actuators per side to replace the 46 muscles per side found in GaitBody2392. Of these, only the four muscles that control toe movement seem to be a good match (Fig. 15). Other muscles need various degrees of refinement. For many, like the gastrocnemius and soleus (Figs. 11 and 12, respectively), the method to improve the design is straightforward. For bipedal robot design, it is important to look at specific muscles, groups of them based on their function, and groups of all muscles around a particular joint.

Tools have been created in Matlab to calculate and visualize values for length, moment arm, force, and torque for the robot design using PAMs. These values can be compared to existing human models. Further refinement of these tools and the robot design are possible. The robot design can be solid modeled and the skeletal structure can be 3D-printed - the initial steps towards building and testing a physical biomimetic bipedal humanoid robot.

References

1. Hosoda, K., Sakaguchi, Y., Takayama, H., & Takuma, T. (2010). Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure. *Autonomous Robots*, *28*(3), 307-316.

2. Liu, X., Rosendo, A., Ikemoto, S., Shimizu, M., & Hosoda, K. (2018). Robotic investigation on effect of stretch reflex and crossed inhibitory response on bipedal hopping. *Journal of The Royal Society Interface*, *15*(140), 20180024.

3. Reher, J., Cousineau, E. A., Hereid, A., Hubicki, C. M., & Ames, A. D. (2016, May). Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1794-1801). IEEE.

4. Radford, N. A., Strawser, P., Hambuchen, K., Mehling, J. S., Verdeyen, W. K., Donnan, A. S., ... & Berka, R. (2015). Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, *32*(3), 397-419.

5. Liu, J., & Urbann, O. (2016). Bipedal walking with dynamic balance that involves three-dimensional upper body motion. *Robotics and Autonomous Systems*, *77*, 39-54.

6. Shin, H., Ikemoto, S., & Hosoda, K. (2018). Constructive understanding and reproduction of functions of gluteus medius by using a musculoskeletal walking robot. *Advanced Robotics*, *32*(4), 202-214.

7. Ishii, Y., Nishikawa, S., Niiyama, R., & Kuniyoshi, Y. (2018, October). Development of a Musculoskeletal Humanoid Robot as a Platform for Biomechanical Research on the Underwater Dolphin Kick. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3285-3291). IEEE.

8. Nakanishi, Y., Ohta, S., Shirai, T., Asano, Y., Kozuki, T., Kakehashi, Y., ... & Urata, J. (2013). Design approach of biologically-inspired musculoskeletal humanoids. *International Journal of Advanced Robotic Systems*, *10*(4), 216.

9. Asano, Y., Kozuki, T., Ookubo, S., Kawamura, M., Nakashima, S., Katayama, T., ... & Kakiuchi, Y. (2016, November). Human mimetic musculoskeletal humanoid Kengoro toward real world physically interactive actions. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*(pp. 876-883). IEEE.

10. Asano, Y., Okada, K., & Inaba, M. (2017). Design principles of a human mimetic humanoid: Humanoid platform to study human intelligence and internal body system. *Science Robotics*, *2*(13), eaaq0899.

11. Chou, C. P., & Hannaford, B. (1996). Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE Transactions on robotics and automation*, *12*(1), 90-102.

12. Klute, G. K., Czerniecki, J. M., & Hannaford, B. (1999, September). McKibben artificial muscles: pneumatic actuators with biomechanical intelligence. In *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No. 99TH8399)* (pp. 221-226). IEEE.

13. Tondu, B., & Zagal, S. D. (2006, February). McKibben artificial muscle can be in accordance with the Hill skeletal muscle model. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.* (pp. 714-720). IEEE.

14. Seth, A., Sherman, M., Reinbolt, J. A., & Delp, S. L. (2011). OpenSim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange. *Procedia Iutam*, *2*, 212-232.

15. Gilroy, A., MacPherson, B., Ross, L., Schünke, M., Schulte, E. & Schumacher, U. (2012). *Atlas of anatomy*. Stuttgart New York: Thieme.

16. Standring, S. (2016). *Gray's anatomy : the anatomical basis of clinical practice*. Philadelphia: Elsevier Limited

17. Thelen, D. G., Seth, A., Anderson, F.C., Delp, S.L.. Gait 2392 and 2354 Models. Standfod University. https://simtk-confluence.stanford.edu/display/OpenSim/Gait+2392+and+2354+Models (last accessed 22 March, 2019).

18. Stredney, D. L. (1982). The representation of anatomical structures through computer animation for scientific, educational and artistic applications.

19. Delp, S. L., Loan, J. P., Hoy, M. G., Zajac, F. E., Topp, E. L., & Rosen, J. M. (1990). An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical engineering*, *37*(8), 757-767.

20. Platzer, Werner (2004). *Color Atlas of Human Anatomy, Vol. 1: Locomotor System* (5th ed.). (pp. 244–246). Thieme.

21. I. A. Kapandji, Physiology of the Joints. Churchill Livingstone, 1986.

22. Hoy, M. G., Zajac, F. E., & Gordon, M. E. (1990). A musculoskeletal model of the human lower extremity: the effect of muscle, tendon, and moment arm on the moment-angle relationship of musculotendon actuators at the hip, knee, and ankle. *Journal of biomechanics*, *23*(2), 157-169.

23. Sherman, M. A., Seth, A., & Delp, S. L. (2013, August). What is a moment arm? Calculating muscle effectiveness in biomechanical models using generalized coordinates. In *Proceedings of the... ASME Design Engineering Technical Conferences. ASME Design Engineering Technical Conferences* (Vol. 2013). NIH Public Access.

24. Festo. Fluidic Muscle DMSP/MAS. https://www.festo.com/cat/en-us_us/data/doc_enus/PDF/US/DMSP_ENUS.PDF (last accessed 22 March, 2019).

25. Millard, M., Uchida, T., Seth, A., & Delp, S. L. (2013). Flexing computational muscle: modeling and simulation of musculotendon dynamics. *Journal of biomechanical engineering*, *135*(2), 021005.

26. Thelen, D. G. (2003). Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *Journal of biomechanical engineering*, *125*(1), 70-77.

27. Hunt, A. J., Graber-Tilton, A., & Quinn, R. D. (2017, August). Modeling length effects of braided pneumatic actuators. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V05AT08A008-V05AT08A008). American Society of Mechanical Engineers.