**Guidance on Using OpenSim Simulation on Ankle Inversions**

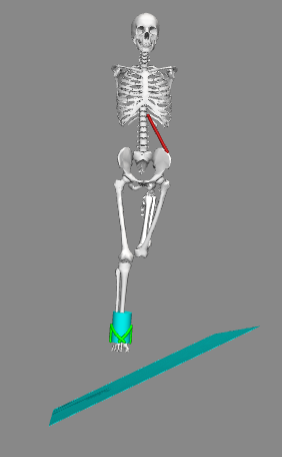
***Zehao’s***

1. **Simulation through GUI of OpenSim**
   1. **Drop Landing Simulation**

The purpose of the drop landing simulation in OpenSim is to predict the risk of ankle inversion injury by free landing onto an inclined plate under different conditions which include: wearing an ankle foot orthosis (AFO) or not wearing an AFO as well as the height and angle of the landing platform.

* + 1. **GUI of Drop Landing Simulation**

Open the file in OpenSim software GUI, the drop landing system is shown which consists of musculoskeletal model, ankle foot orthosis and landing platform.



Musculoskeletal Model

Landing Platform

Ankle Foot Orthosis

**Figure 1:** Illustration of drop landing model consisting of musculoskeletal model, ankle foot orthosis and landing platform

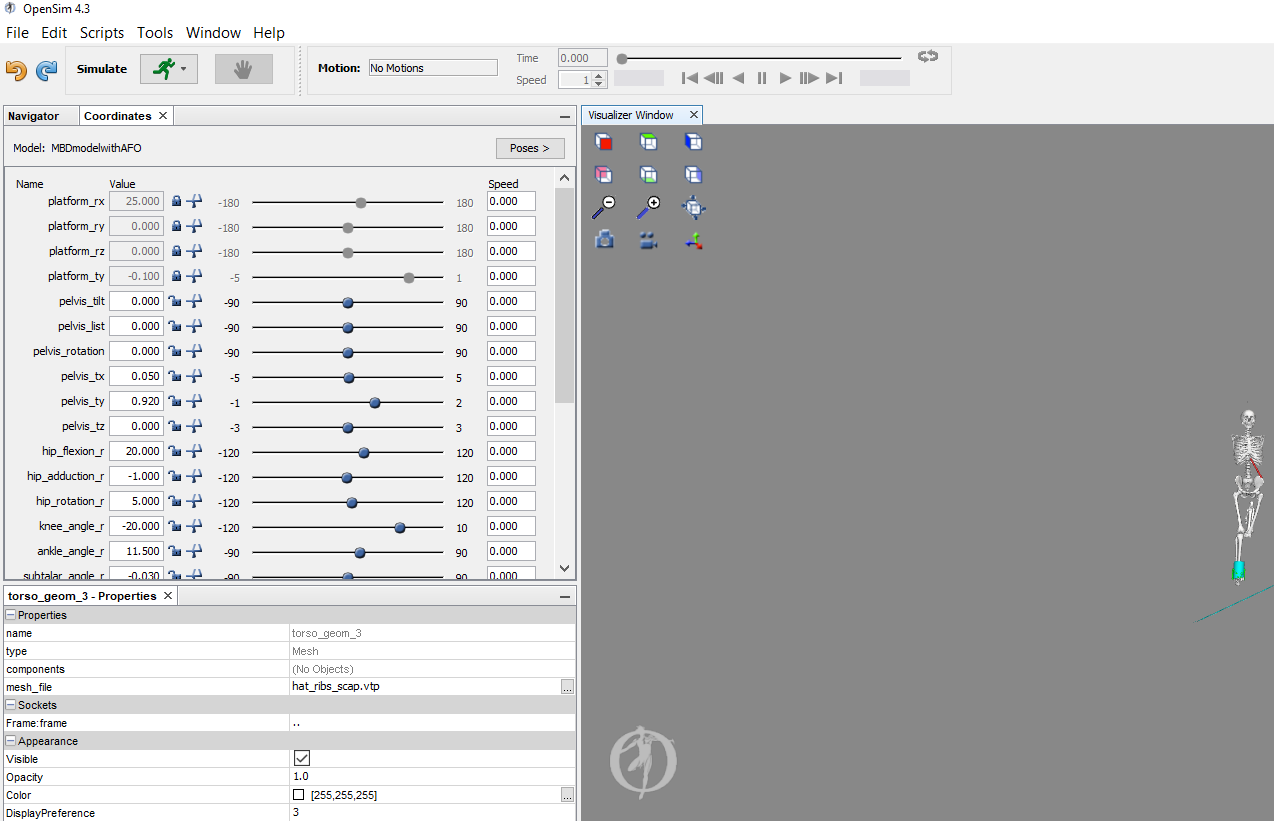
All of these three conditions can be adjusted in the software in order to predict the ankle inversion during a drop landing in different situations.

**Explore the Musculoskeletal Model**

The musculoskeletal model used in Drop Landing Simulation has a torso, a pelvis and two legs with a total of 23 degrees of freedom and 70 muscle-tendon actuators. Various components of the model can be explored using the **Navigator** panel including **Bodies** and **Joints**. To evaluate the level of ankle inversion, motion between the tibia and right foot is focussed which is associated with the subtalar joint in the ankle. The subtalar joint of right ankle can be found as **subtalar\_r** in the **Joints** group in the Navigator panel.

**Landing Platform**

The height of drop and inclined angle of landing platform will influence the level of ankle inversion. These can be adjusted in the **Coordinates** tab in the GUI where **platform\_rx** represents the pre-defined slope angle of the landing platform and drop height can be worked out by calculating the difference between **pelvis\_ty** (pre-defined height of musculoskeletal model) and **platform\_ty** (pre-defined height of the landing platform). The angle and position of other sections in the musculoskeletal model such as knee or hip can also be adjusted in the ‘Coordinate’ tab if needed.

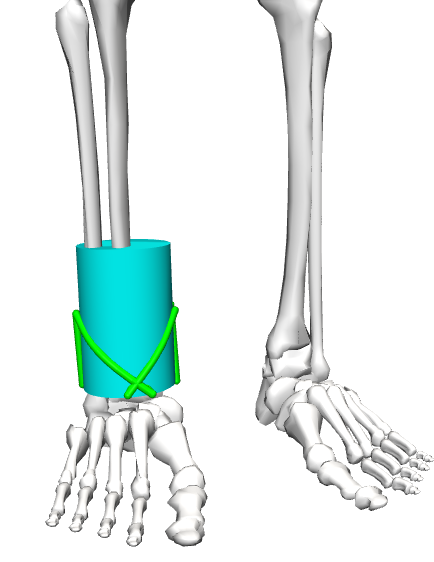
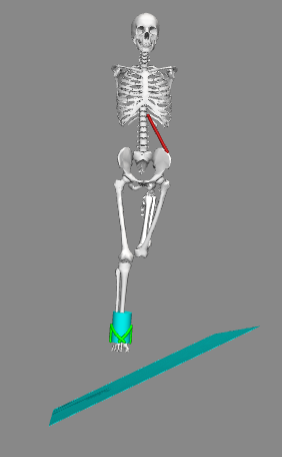


Coordinate Tab

**Figure 2:** Illustration of Coordinate Tab in OpenSim software GUI

**Ankle Foot Orthosis**

An ankle foot orthosis (AFO) is added on the right foot ankle of the musculoskeletal model in order to evaluate the effect of wearing an ankle foot orthosis to ankle inversion in drop landing. In this example, the AFO design is constructed with four straps which is simulated to the AFO prototype utilized in real life experiments. A cylinder (in green colour as shown in Figure 3) is also developed to represent human leg and the AFO straps are wrapped around the cylinder in the model.



Strap 1

Strap 2

Strap 3

Strap 4



**AFO Prototype**

**AFO Design in Simulation**

Cylinder that represents human leg

**Figure 3:** Illustration of AFO design used in the drop landing model which consists of four straps and the actual AFO prototype utilized in experiments

Text, letter

Description automatically generatedThe four straps of AFO, Strap 1, Strap 2, Strap 3 and Strap 4, were simulated as ligaments in the OpenSim model, and can be found as **orthosis\_1**, **orthosis\_2**, **orthosis\_3** and **orthosis\_4** in the Navigator tab respectively. To fully evaluate the influence of different AFO designs on the ankle joint motion during drop landing simulation, various design variables of AFO can be changed in the OpenSim model including: mechanical properties of each AFO strap as well as number, orientations and locations of AFO straps. All of these design variables can be varied through modifying the parameters in the **AFO input text file**.

**Figure 4:** Design variables of Ankle Foot Orthosis model in the AFO input text file.

Number, Orientation and Location of AFO Straps

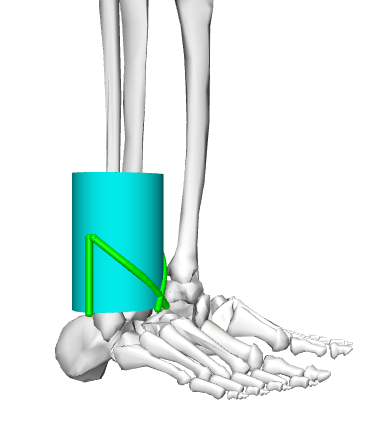
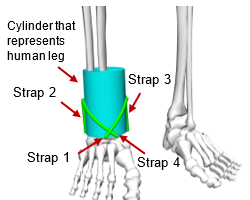
A picture containing text

Description automatically generatedThe number of straps of the AFO model can be varied through changing the AFO text input file. In this example, a total of four straps is utilized and each strap is numbered. The design variables related to the orientation and location of AFO straps are highlighted in the image below.

**Figure 5:** Design variables included in the red rectangle are utilized to vary the orientation and location of AFO straps

**AFO\_bottom\_location** defines the origin coordinate of AFO model and the location and orientation of other AFO straps will use this coordinate as reference.

**AFO\_cylinder\_radius** defines the radius of cylinder which represents to the human leg.



Strap 2

**Figure 6:** Representation of orientation and location of four AFO straps

**AFO\_height** defines the length of AFO straps that are placed vertically, such as Strap 2 and Strap 3 in this example.

**AFO\_bottom\_location\_angle** is utilized to define the bottom location of each AFO strap by setting the angle between the x-axis (global coordinate axes of the musculoskeletal model) and the bottom point of the AFO strap. In this example, the four elements in the AFO\_bottom\_location\_angle list represent the angle of bottom location of AFO Strap 1, Strap 2, Strap 3 and Strap 4, respectively. An example is illustrated in Figure 7 where point O is the origin point, point B is the bottom location point of an AFO strap, the bottom location angle of this AFO strap is defined as , and r is the pre-defined radius of the cylinder (human leg). Afterwards, the coordinate of point B in the global coordinate system can be calculated.



**Figure 7:**  Schematical example showing how to define the bottom location of AFO straps in OpenSim

**AFO\_Strap\_Orientations** is adopted to define the orientation angle of an AFO strap compared to z-axis (vertical axis) in the global coordinate system of the model. Similarly, the AFO\_Strap\_Orientations list has four elements in this example which represent the orientation angle of Strap 1, Strap 2, Strap 3 and Strap 4, respectively. Because the Strap 2 and 3 are placed vertically in the current example, their orientation angle is . While Strap 1 and 4 are designed to cross each other, and thus they have an orientation angle of and , respectively. A visual example is given in Figure 8 where the line connecting point B and T representing Strap 4 of the AFO. The orientation angle of this AFO strap is defined as in the diagram. Afterwards, both the coordinate of point T and actual length of Strap 4 can also be calculated.



Height of AFO



B

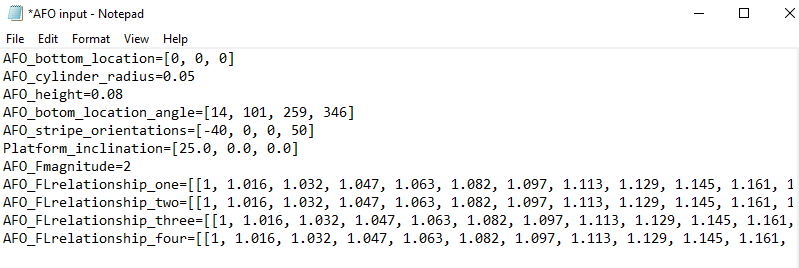
T

**Figure 8:** Diagram presenting the orientation angle of AFO Strap 4 and calculations of its length in this example

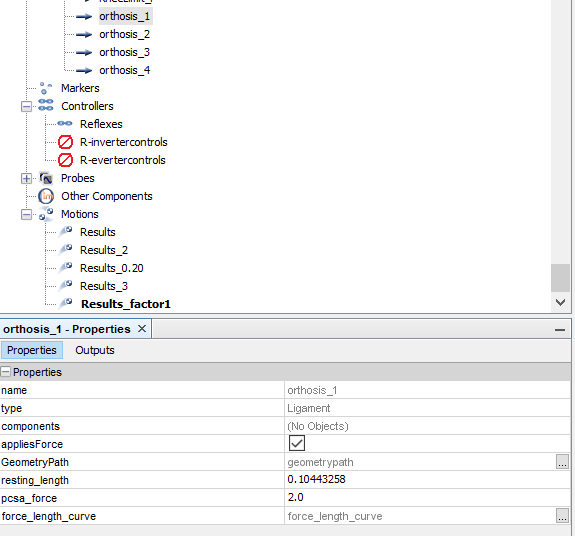
**Platform\_inclination** apart from setting in OpenSim software GUI, the incline angle of landing platform can also be set in the AFO Input text file.

Mechanical Properties of each AFO Strap

Design variables utilized to adjust the mechanical properties of each AFO strap are presented in Figure 9. These variables can also be defined in the OpenSim software GUI by finding the particular AFO strap in the Navigation tab (Figure 10).



**Figure 9:** Design variables included in the red rectangle are utilized to vary the mechanical properties of AFO straps



Select the AFO Strap being defined

Length of the AFO Strap

Mechanical Properties of the AFO Strap

**Figure 10:** Diagram showing how to adjust the design variables for defining mechanical properties of AFO Strap 1

**AFO\_Fmagnitude** is an amplification factor used for amplifying the force provided by AFO straps. This factor can also be defined in the OpenSim software GUI as **psca\_force** as shown in Figure 10.

**AFO\_Flrelationship** is utilized to define the force\_length relationship of the AFO Strap. The input force\_length relationship curve is according to actual mechanical behaviour of material utilized for the AFO strap. The x-axis of the curve is calculated as dividing the length of AFO strap after extension by the original length of the strap,

where is the original length of the AFO strap, is length of the strap after extension. While y-axis of the curve is the force exhibited by the AFO strap at different extension length. The x-axis and y-axis values of the force\_length relationship curve can be input by either AFO Input text file shown in Figure 10 or OpenSim software GUI, and the input curve can be viewed in the GUI (Figure 11).

Chart, line chart

Description automatically generated

**Figure 11:** Input Force\_length relationship curve of AFO Strap 1

**1.2.1 Perform the Drop Landing Simulation**

After all the above parameters are set, drop landing simulation can be performed in the OpenSim GUI by clicking the simulate button (Figure 12). The time period of simulation can be defined as well.

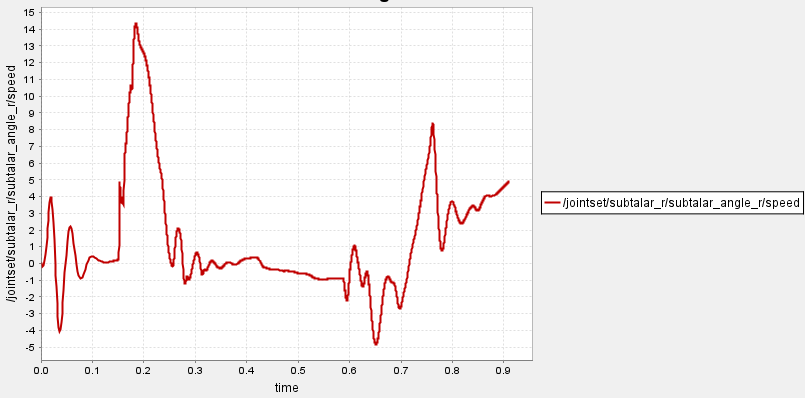
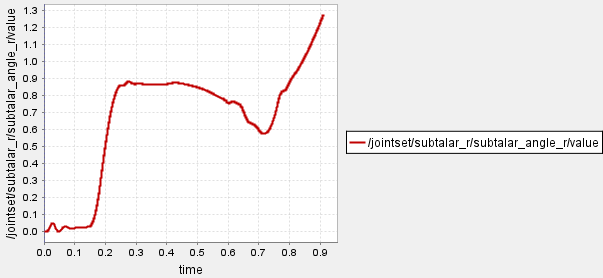
Graphical user interface, application, Word

Description automatically generated

**Figure 12:** Simulate button to perform the drop landing simulation

**1.2.2 Plot the Simulation Results**

To plot the simulation results, a new plot window can be opened by selecting **Tools>Plot** in the OpenSim GUI. Afterwards, click the **Y-Quantity**  button and select **Load file** to select the set of result being plot. In the **Filter by pattern** text box, search for the results wanted to be plot. For evaluating the level of ankle inversion, **subtalar\_angle\_r/value** (radians)and **subtalar\_angle\_r/speed** (radians/second)**,** representing the rotation angle and angular speed of subtalar joint of right foot ankle during drop landing simulation, are selected to plot. For the x-axis, click the **X-Quantity** button and choose time. Afterwards, plots illustrating rotation angle of subtalar joint against time and angular speed of subtalar joint against time during drop landing simulation can be generated. Examples of these plots are shown in Figure 13.



**(a)**

**(b)**

**Figure 13:** Examples of (a) Rotation angle of subtalar joint against time (b) Angular speed of subtalar joint against time during drop landing

Typically, for evaluating the motion of ankle inversion during drop landing, only the results within the beginning 0.25 second of simulation is analysed.

***Xijin Hua’s***

The batch simulation mainly comprises of three parts: (1) the MSK models for drop landing, walk and running, (2) the AFO representation developed in the MSK models, (3) the batch simulation.

1. **MSK model**
2. ***Drop landing model:***

***Origin model:*** The drop landing model was developed based on Delp’s drop landing model (Reference: Preparatory co-activation of the ankle muscles may prevent ankle inversion injuries), the model can be downloaded from the GitHub (e.g. Origin model/drop landing) or (https://simtk.org/projects/ankle-sprains).

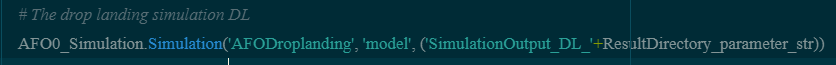
***GUI model running:***The model was running using a forward dynamics analysis, which can be performed in the OpenSim GUI by clicking the simulation buttonGraphical user interface, application, Word

Description automatically generated. The time for the simulation is usually set as 0.4 s or 0.5 s (before falling down).

***AFO model:***The AFO drop landing model is based on the original model, with making some changes on the model: (1) the first change is to remove the upper limbs and add the weight of the upper limbs to the torso segment. To this end, open the MSK model (\*.osim file) using a text editor (e.g. notepad++), and delete the related text of the upper limb bodies (i.e. humerus, ulna, radius, hand) from the text; (2) the second change is to add a wrap cylinder on the right tibia, which represents the foot segment and acts as wrap surface for the AFO representation. The wrap cylinder is added to the MSK model by adding a “WrapCylinder” class manually in the model text.

The AFO representation will be then added to the MSK model as detailed in section 2. The AFO representations in the MSK model are modelled as ligament elements, which are added to the MSK model by adding the Ligament Class related text to the MSK model text, using the python module (AFO2\_MBDModel.py). The parameters for the Ligament element, in terms of the resting\_length, pcsa\_force and force\_length\_curve are able to automatically change in the code. The initial value of these parameters are stored in the AFO design.txt file in the AFO design folder, which can also be changed as needed.

***AFO model display and simulation in the code:*** In the batch simulation code, the model can be displayed using the syntax:



The model can also be run by using ‘simulation’ to replace ‘model’ in the syntax:



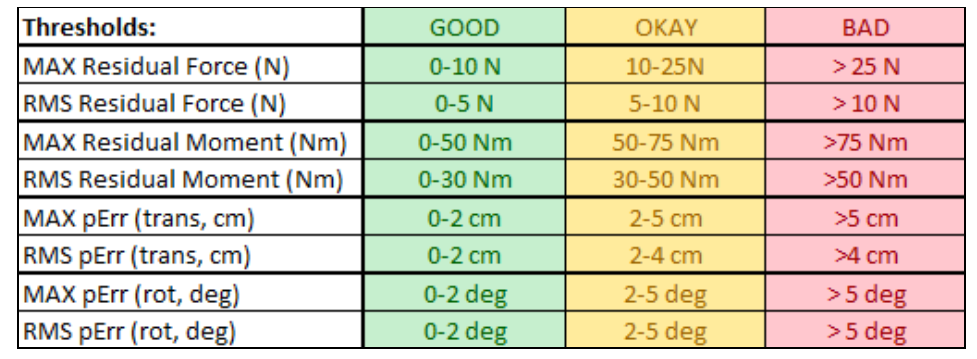
1. ***Walk and running model:***

***Origin model:*** The model for the walk and running simulation was developed based on Rajagopal full body model (Reference: Full-Body Musculoskeletal Model for Muscle-Driven Simulation of Human Gait). The model can be downloaded from the GitHub (e.g. Origin model/walk and running) or (https://simtk.org/projects/full\_body/).

***AFO gait model:*** The MSK model for walk and running is based on the Rajapopal model, with some changes: (1) the first change is to remove the upper limb and add the weight of the upper limb to the torso segment, this can be done by removing the upper limb related text in the model text and changing the weight of torso with adding the weights of the upper limb; (2) the second change is to enable the subtalar\_angle\_r and subtalar \_angle\_l coordinates. This can be done by unlocking the coordinates using  from the main menu Window – Coordinates, or changing the text (<locked>true</locked>) from “true” to “false” under the class of “subtalar\_angle\_r” and “subtalar\_angle\_l” in the model text; (3) the third change is to add a wrap cylinder on the right tibia, which represents the foot segment and acts as wrap surface for the AFO representation. The wrap cylinder is added to the MSK model by adding a “WrapCylinder” class in the model text. This, hower, should be done through adding a “WrapCylinder” class in the model text manually after the scaling of the model (e.g. “Fullbodymodel\_Walk\_RRA\_adjusted.osim” located in: Simulation\_printAFO/Gait simulation/Model outputs/3\_RRA/ Fullbodymodel\_Walk\_RRA\_adjusted.osim).

***GUI model running:*** The procedures for the simulation of walk and running include five steps: model scaling, inverse kinematics (IK), residual reduction algorithm (RRA), computed muscle control (CMC), and forward dynamics (FD). Each step requires setup file, which is included in the folder, locating in: Simulation\_printAFO/Gait simulation/Setup files.

1. *Model scaling*: This process is to scale the generic model to a subject specific model through markers. To do this, open the MSK full body model (Simulation\_printAFO\_CAMG/Gait simulation/Fullbodymodel\_gait.osim), in the main menu, choose Tools – Scale Model, in the Scale Tool, load the scale tool setup file (located in: Simulation\_printAFO/Gait simulation/Setup files/1\_Walk\_Scale\_setup.xml) and run. The model will be scaled to match the anthropometric features of the subject, the scaled model will be located in: Simulation\_printAFO/Gait simulation/Model outputs/1\_Scale/Fullbodymodel\_Walk\_Scale.osim. At the same time, the scaled model will be also displayed in the OpenSim main page.
2. *Inverse kinematics*: after the scaling, an inverse kinematics will be performed. Make the scaled model as the current model (by write clicking the scaled model and choose “make current”), in the main menu, choose Tools – Inverse Kinematics, in the Inverse Kinematics Tool, load the IK setup file (located in: Simulation\_printAFO/Gait simulation/Setup files/2\_Walk\_IK\_setup.xml) and run. The inverse kinematics analysis will be performed. The IK result will be located in: Simulation\_printAFO/Gait simulation/Model outputs/2\_IK/Walk\_IK\_outputmotion.mot.
3. *Residual reduction algorithm* (RRA): the RRA is used to reduce the dynamics inconsistency between the measured kinematics and ground reaction force data. The RRA should be run several times until the residual force, moment and body position reached the thresholds recommended below.



To perform RRA, from the main menu, choose Tools – Reduce Residuals, in the RRA Tool, load the RRA setup file (located in: Simulation\_printAFO/Gait simulation/Setup files/3\_Walk\_rra\_setup\_rra1.xml) and run. After RRA, change the mass of each body and save, and repeat the process until the error thresholds achieved.

After each RRA, the RRA tool will suggest a total mass change to the model, and will distribute that mass change over each body in the model (which can be found in the message windows in the OpenSim). The mass of each body should be then changed manually in the model. This can be done by selecting the body and choosing Windows – Properties, in the properties tool, change the mass as suggested. Repeat the RRA step until the above thresholds achieved. It should be noted that for the second RRA, should load the “3\_Walk\_rra\_setup\_rra2.xml” setup file in the RRA Tool.

After completing RRA, the optimal forces for all muscle-tendon units should be adjusted using the scaled model’s mass and height. This can be done by running the following code:

------------------------------------------------

import opensim

import RRAModelMassModification

osimModel="HomeDir\Simulation\_printAFO\Gait simulation\Model outputs\\3\_RRA\Fullbodymodel\_Walk\_RRA\_modification\_final.osim"

originModel=opensim.Model(osimModel)

osimModel\_adjusted= RRAModelMassModification.ScaleOptimalForceSubjectSpecific(originModel, originModel, 1.7, 1.83)

osimModel\_adjusted= RRAModelMassModification.setMaxContractionVelocityAllMuscles(osimModel\_adjusted, 15)

osimModel\_postRRA.printToXML("HomeDir\Simulation\_printAFO\Gait simulation\Model outputs\\3\_RRA\Fullbodymodel\_Walk\_RRA\_adjusted.osim")

------------------------------------------------

Where the HomeDir is the folder where the “Simulation\_printAFO” is put in.

After running the code, an adjusted model will be created, which will be located in: Simulation\_printAFO\Gait simulation\Model outputs\3\_RRA.

1. *Computed muscle control* (CMC): the CMC will compute the muscle excitation levels that drive the generalized coordinates of model towards desired kinematic trajectories. To perform CMC, open the adjusted model “Fullbodymodel\_Walk\_RRA\_adjusted.osim” (located in: Simulation\_printAFO\Gait simulation\Model outputs\\3\_RRA), from the main menu, choose Tools – Computed Muscle Control, in the CMC Tool, load the CMC setup file (located in: Simulation\_printAFO/Gait simulation/Setup files/4\_Walk\_cmc\_setup.xml) and run. After CMC, the result will be located in: Simulation\_printAFO/Gait simulation/Model outputs/4\_CMC).
2. *Forward dynamics (FD):* the FDwill predict the motion that is generated from the muscle activations calculated in CMC. To perform FD, make the adjusted model as the current model, from the main menu, choose Tools – Forward Dynamics, in the Forward Dynamics Tool, load the FD setup file (Simulation\_printAFO/Gait simulation/Setup files/ 5\_Walk\_Forward\_setup\_withoutAFO.xml) and run. After FD, the results will be located in: Simulation\_printAFO/Gait simulation/Model outputs/ 5\_ForwardDynamics.

The processes for the running are the same as walk.

***Walk and Run simulation through code:*** The whole processes of the walk and running simulation can be performed using the following code:

------------------------------------------------

import AFO0\_Simulation

import os

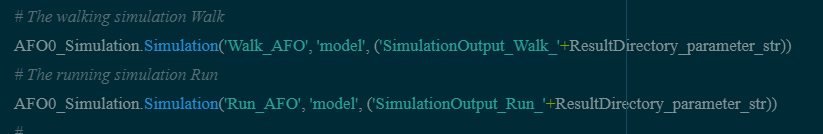
AFO0\_Simulation.Simulation('walk', 'simulation', results\_directory='')

AFO0\_Simulation.Simulation('run', 'simulation', results\_directory='')

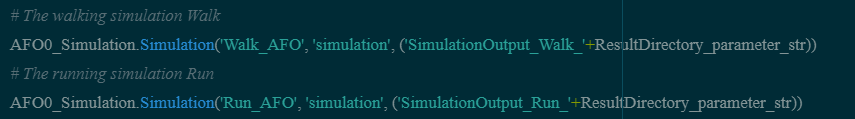
------------------------------------------------

The results of the simulation will be located in: Simulation\_printAFO/Gait simulation/Model outputs/, which will include 1\_Scale, 2\_IK, 3\_RRA, 4\_CMC, and 5\_ForwardDynamics subfolders.

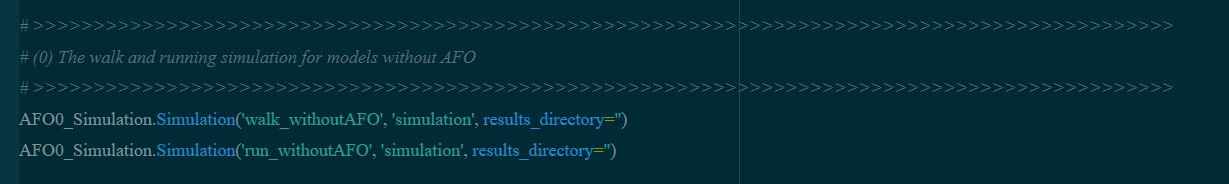
***AFO model display and simulation in the code:*** In the batch simulation code, the model can be displayed using the following syntax:



The model can also be run by using ‘simulation’ to replace ‘model’ in the syntax:



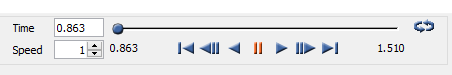
These codes were used to run the simulation for the models with AFO, for the simulations without AFO, should use a separate code:



This just needs to be run one time before the batch simulation.

1. **The collection of the simulation results**

The simulation results are stored in: Simulation\_printAFO \Gait simulation\Model outputs, after each step of walk simulation. There are two ways to display the simulation: one is to use OpenSim GUI and the other one is to use excel.

1. *Using OpenSim GUI:* For example, if want to display and check the subtalar angle of the right foot after CMC, firstly to open the AFO model (Simulation\_printAFO\Gait simulation\Model outputs\3\_RRA\Fullbodymodel\_Walk\_RRA\_final\_AFO), load the motion through File – Load Motion, and navigate to: Model outputs\4\_CMC\cmc\_states.sto, and then choose Open. The motion is loaded. You can check the animation of the movement through . To display the subtalar angle, in the main menu, choose Tools – Plot, in the Plotter tool, click the **Y-Quantity**  button and select **Load file** to select the set of result being plot (e.g., in this case, select cmc\_states). In the **Filter by pattern** text box, search for the results wanted to be plot, e.g. subtlar\_angle, tick the selected box, and click OK. For the x-axis, click the **X-Quantity** button and choose cmc\_states, click OK, and to plot the subtalar angle, click Add. The results will be displayed in the Plotter tool.
2. *Using excel:*  you can also process and analyse the data through excel. There are many results stored in the results fold, such as Kinematics, Muscle forces, etc. Just simply drag the related results to the excel, and process the results in the excel.
3. **The representation of AFO in the musculoskeletal model**

The 3D printed AFO will be represented as several nonlinear straps, modelling as ligament elements in the musculoskeletal (MSK) model, as shown in Figure 1:



Figure 1: The 3D printed AFO in the musculoskeletal model was represented as several straps

1. ***The locations and geometries of the AFO representation***

The location of the AFO strap in the MSK model, the size and nonlinear mechanical properties of the straps were determined using a series of design variables, which were summarized in a text file (e.g. AFO input.txt), as shown in Figure 2. The AFO representation in the MSK model was then generated based on these design variables. The details about how to generate the AFO representation will be detailed below.

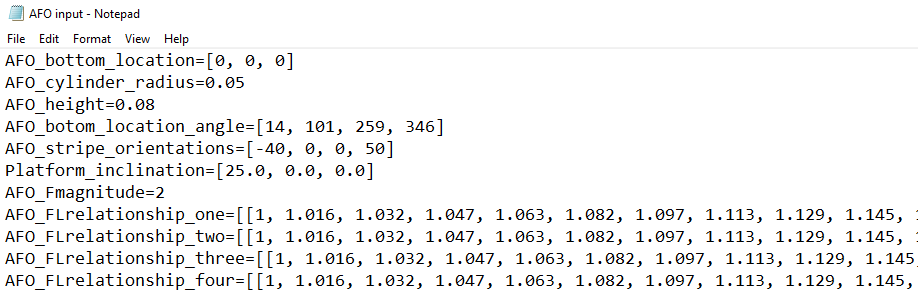


Figure 2: the design variables for the AFO representation in the MSK model

***AFO\_bottom\_location:*** this variable defines the location of the centre of the AFO bottom relative to the talus bone. The default values of [0, 0, 0] means that the centre of the AFO bottom is located at the geometry centre of the talus bone in the MSK model.

***AFO\_cylinder\_radius:*** this variable defines the radius of the cylinder, which represents the size of the leg of the subject, on which the AFO will wrap. Therefore, this radii will also determine the radii of the circles for the top and bottom of the AFO stripes.

***AFO\_height:*** this variable defines the height of the AFO along the leg segment of the subject.

***AFO\_bottom\_location\_angle:*** this variable defines the locations of the endpoints of the AFO representation at the bottom. These endpoints are created along the circumferences of the circle at the bottom of the AFO cylinder, the locations of which are therefore defined using the central angles of these points, as shown below.



Figure 3: The definition of the bottom endpoints of the AFO stripes in the MSK model

Once the locations of the bottom endpoints of the AFO stripes are defined in terms of central engles, their local coordinates in the model global coordinate system can be determined using the following equations:



Where (XB, YB, ZB) is the local coordinate of one endpoint B at the bottom of the AFO, (X0, Y0, Z0) is the local coordinates of the centre of the circle at the AFO bottom (e.g. AFO\_bottom\_location). r is the radius of the cricle and θ is the centre angles of the endpoints (e.g. the AFO\_bottom\_location\_angle). The X axis is defined as the direction from the heel to the toe, while the Z direction is defined as direction from the medial and lateral. The positive θ is a clockwise rotation when looking from the knee to the foot.

***AFO\_stripe\_orientations:*** this variables defines the orientations of the AFO stripes, which is defined as the angle between the AFO stripe and the vertical direction when the AFO cylinder is unfold, as shown in below:



Figure 4: The definition of the stripe orientations in (a) 3D AFO cylinder and (b) the AFO cylinder when it is unfold

Once the orientations of the AFO stripes are defined, the locations of the endpoints of the stripes at the top of AFO can be calculated:

As shown in Figure 4a, length of curve TB’ = θ’/2π\*2πr = θ’\*r

In Figure 4b, length of TB’ = h \* tan ϕ

Therefore, the θ’ can be obtained as θ’ = h\*tan ϕ/r

The locations of the endpoints of the AFO stripes at the top of the AFO (T) can be then calculated as:



where (XT, YT, ZT) is the local coordinate of the endpoint T at the top of the AFO, (X0, Y0, Z0) is the local coordinates of the centre of the circle at the AFO bottom (e.g. AFO\_bottom\_location). r is the radius of the cricle, h is the height of the AFO, and θ is the centre angles of the endpoint B at the bottom (e.g. the AFO\_bottom\_location\_angle), and . ϕ is the orientation of the stripe.

The AFO stripe representation in the MSK model is modelled as ligament element, with the geometry defined as the 3D curve connecting the paired endpoints at the top and bottom of the AFO (e.g. the endpoints T and B) and wrapping the AFO cylinder. The slack length of the AFO stripe can be calculated as:

Lslack = h/cos ϕ

***Platform\_inclination:*** this variable defines the platform inclination angles for the drop landing simulation in three directions. By combing the angles in three directions, any inclination angles for the platform can be achieved.

1. ***The mechanical properties of the AFO materials***

The AFO stripes in the MSK model are modelled as nonlinear ligament elements, the nonlinear mechanical properties of which are characterised using a force-length relationship curve and a force magnitude, as shown in the Figure 4.



Figure 5: The force-length curve defined for the AFO stripe in the MSK model

***AFO\_Fmagnitude***: this design variable defines the force magnitude generated by the AFO stripe, the actual force generated by the AFO stripe during the simulation is determined by the AFO force magnitude multiplying the force length curve. For example, as shown in Figure 5, the force generated by the AFO stripe with an extension rate of 1.6 is about 1.16 N (e.g. 2N x 0.58).

***AFO\_FLrelationship\_one (\_two, \_three, \_four):*** These design variables define the nonlinear mechanical properties of the AFO stripes in terms of force-length relationship, as shown in Figure 5. This force-length relationship is composed of two vectors: the first vector (x values in Figure 5) is the extension rate of the AFO stripes during the simulation, which is defined as:

Extension rate x = L1 / L0

where L1 is the length of the AFO stripe after extension (stretch length), L0 is the original length before extension (slack length); the section vector (y values) is the force generated by the AFO stripe, however, the actual force generated by the strip should be the y values times the AFO\_Fmagnitude.

1. **Design variables for the mechanical properties of AFO strips**

The design variables of the locations for the bottom endpoints of the AFO strips, and the orientations of the AFO strips are straightforward.

For the mechanical properties of the AFO materials, we defined two design variables: one is the amplification of the force-length curve (i.e. fl\_am\_\* in the main code), the other one is the shift of the force-length curve (i.e. fl\_shift\_\* in the main code).

The amplification of the force-length curve represents the scaling of the force generated by the AFO strips, indicating the number of the fibres in each AFO strip, as shown in Figure 6a. For examples, if the baseline AFO material can produce a force of 0.6N at a strain of 0.4, then the new materials with a scaling of 60 (design variables of fl\_am\_\* times the FL\_amplification\_stepsize in the main code) can produce a force of 36N at the same strain of 0.4.

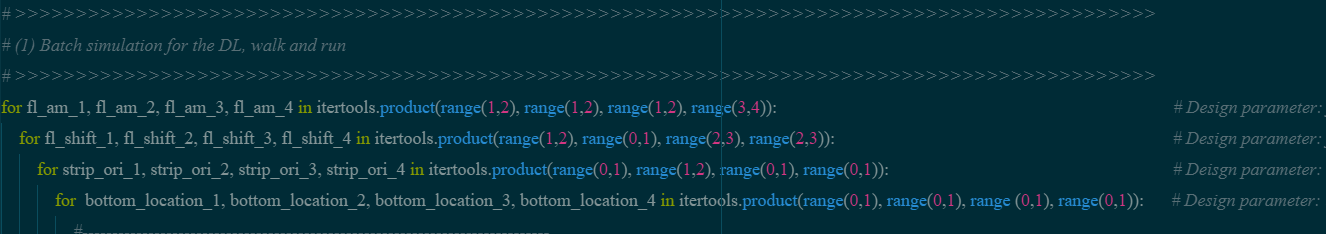
Similarly, the shift of the force-length curve represents the translation of the curve as shown in Figure 6b. For example, the shift value = -0.2 (is calculated by fl\_shift\_\* times FL\_shift\_stepsize -0.2 in the main code) means the curve is translated 0.2 units to the left. That means, for example, if the baseline AFO strip produce a force of 0.6N at a strain of 0.4, to produce the same force of 0.6N, the strain in the translated curve is 0.2 (0.4-0.2).



Figure 6: (a) the design variable of amplification for the mechanical properties of AFO materials, representing the scaling of the force (Blue: the baseline properties, Orange: the properties after scaling); (b) the design variable of shift for the mechanical properties of the AFO materials, representing the shift of the curve (Bule: the baseline properties, Orange: the curve after shifting to left).

1. **Batch simulation for optimization**

The batch simulation for the drop landing, walk and run can be performed through a loop process, as shown below. The detailed explanation of the batch simulation codes can be found in the readme file in the github.



The process of the batch simulation is as following:

1. Give an initial guess (trial) of the design variable of the AFO in the AFO design.txt file which is located in: Simulation\_printAFO\AFO Design (the design variables can be changed here);
2. Decide the ranges of the parameters in the loop code (e.g. “for fl\_am\_1, fl\_am\_2, fl\_am\_3, fl\_am\_4 in itertools.product(range(2,4), range(8,9), range(1,2), range(3,4)):”), the design parameters of the AFO will be changed automatically in the MSK model;
3. Run the batch simulation and collect the results of interests (e.g. subtalar angles, ankle angles, muscle activations);
4. Pass the results of interests to the optimization pipeline.