

SIMULATION LAB #2:

Kinematic and Geometric Modeling of the Hip, Knee, and Associated Muscles

Modeling and Simulation of Human Movement
BME 599

Laboratory Developers: Jeff Reinbolt, Hoa Hoang, B.J. Fregly,
Silvia Blemker, Allison Arnold

I. Introduction

Quantitative descriptions of musculoskeletal geometry are an essential building block for developing muscle-actuated, forward dynamic simulations of movement. The path geometry of a muscle determines its moment arms, and therefore, its moment-generating capacity for a given level of muscle force. The force that a muscle can produce also depends on its moment arm — the moment arm determines the length change of the muscle with joint rotation, and the muscle's force-generating capacity depends on its shortening velocity and length. Hence, if descriptions of the musculoskeletal geometry are not sufficiently representative, analyses based on dynamic simulations may be misleading (or perhaps totally bogus...). In this lab, you will gain experience building and evaluating models of musculoskeletal geometry.

II. Objectives

In this simulation exercise you will use OpenSim (Delp, 2007) to:

- Visualize the three-dimensional (3D) surface geometry of the pelvis, femur, tibia, fibula, and selected lower limb muscles
- Implement a simple kinematic description of the hip joint
- Implement simple and complex kinematic descriptions of the knee joint
- Represent lines of action of the semitendinosus muscle over a range of hip and knee joint angles using via points and wrapping surfaces
- Assess the accuracy of your musculoskeletal model by comparing hip and knee flexion moment arms computed with the model to the muscle moment arms determined experimentally from tendon excursion measurements

III. Deliverables

At the completion of this lab, you will need to turn in computer files from your modeling work. Please turn in:

1. A written report

- A Microsoft Word template for the report called **Simulation_Lab2_Report.docx** is available on the course web site.

2. OpenSim model files you created

- **your_name_knee2_muscle1.osim** Your model file with the modified knee #2 along with muscle #1
- **your_name_knee2_muscle2.osim** Your model file with the modified knee #2 along with muscle #2
- **your_name_knee1_muscle3.osim** Your model file with the modified knee #1 along with muscle #3

IV. Input Files

All necessary files for the lab will be provided in **BME599_lab_2.zip** and is available on the course web site.

- **Simulation_Lab2.osim** A base OpenSim model file to create your lower extremity model
- **Simulation_Lab2_Visualize_Anatomy.osim** An OpenSim model file that displays the 3D surface representations of bones and muscles obtained from MR images of a lower extremity cadaver specimen
- **Simulation_Lab2_Arm_Example.osim** A sample file that describes a model of an arm including the joints and muscle paths
- **Simulation_Lab2_experimental_moment_arm.xlsx** Excel file that has the muscle moment arms (of semitendinosus hip extension and knee flexion) determined experimentally on the cadaver specimen
- **Geometry** Folder containing all of the necessary bones in vtp format. VTP is the format that OpenSim uses to represent the 3D bone and muscle surfaces as polygon meshes. Note that fibula_knee1.vtp and tibia_knee1.vtp can be used to construct knee #1 joint. Similarly, fibula_knee2.vtp and tibia_knee2.vtp can be used to construct knee # 2 joint.
- **getRPY.m** A MATLAB m-file that calculates roll, pitch, yaw angles from a transformation (or rotation) matrix

V. Getting Started

There are OpenSim tutorials and an OpenSim manual available for reference. A PDF version of the manual is posted on the course web site. From the class website, download **BME599_Lab_2.zip** to the directory where you are working and unzip the file.

VI. Examine a Sample Kinematic Model of the Arm

In this section, you will load a sample kinematic model of the arm (Murray *et al.*, 1995) and study the corresponding osim file. This model was originally constructed in SIMM and has been converted to OpenSim.

- Select **File → Open Model...** from the OpenSim main menu bar to load the arm model **Simulation_Lab2_Arm_Example.osim**
- In addition, open **Simulation_Lab2_Arm_Example.osim** in a text editor (it is recommended to use Notepad++ or another xml editor to edit OpenSim files so that the file structure is easier to identify)

You will notice in the OpenSim view window that there is a 3D model of a right arm. In the Navigator window, there is the structure of the model including the bodies, joints, and muscles. There are no constraints, contact geometry, or markers in this model, but you may see this in subsequent labs or may need to add them for your models. You can right click on the individual objects to view its properties. You will construct a similar model of the lower extremity, so you may want to use the arm model as a reference while you work through the assignment. Feel free to examine the model and the corresponding joint and muscle structure in the **Simulation_Lab2_Arm_Example.osim** file in a text editor. It may seem confusing at first, but most find it preferable to editing model files in a text editor. You will find that the OpenSim graphical user interface (GUI) is limited when editing models (see **Edit → File (.osim, .xml)...** from the OpenSim main menu bar).

When you are done, close the model by right clicking on the model in the Navigator window and selecting “close.”

VII. Create a Kinematic Model of the Lower Extremity

You will create a kinematic musculoskeletal model that can estimate the lengths and moment arms of the semitendinosus muscle (one of the medial hamstrings) through a range of hip and knee joint angles based on MR images of a cadaver specimen at one body position (the “scanned” position). To do this, you will define reference frames for the pelvis, femur, and tibia, implement kinematic representations of the hip and knee joints, specify attachments of the semitendinosus muscle, and prescribe wrapping surfaces and via points that simulate interactions between the muscle and surrounding structures.

VII-A Visualize Lower Limb Muscle and Bone Surface Geometry

In this section, you will visualize the 3D surfaces of the pelvis, femur, tibia, fibula, and selected muscles that you will use to build your kinematic model. We obtained the 3D surface data by (1) acquiring five series of axial and sagittal MR images, (2) identifying and outlining the structures of interest on each 2D image, (3) generating 3D surfaces of each structure based on the 2D outlines, and (4) registering the surfaces from an adjacent series of images (see Arnold *et al.*, 2000 for details). We have made a simple

OpenSim file called **Simulation_Lab2_Visualize_Anatomy.osim** that displays the 3D bone and muscle surfaces.

- Select **File → Open Model...** from the OpenSim main menu bar to load the model **Simulation_Lab2_Visualize_Anatomy.osim**
- In addition, open **Simulation_Lab2_Visualize_Anatomy.osim** in a text or xml editor

There are a lot of structures on here, but you will not necessarily use them all in your model.

VII-A.1 Examine the 3D anatomy of your lower limb specimen.

Using the Navigator and View windows, identify and examine the pelvis, femur, tibia, and fibula bones, and the medial hamstrings (semimembranosus and semitendinosus) muscles. You can hide all of the other bodies by **right clicking** on the object in the **Navigator** window and selecting **Display → Hide**. You will base your model on these surfaces.

Display the medial hamstrings in wireframe mode by **right clicking** on the object and selecting **Display → Wireframe**.

NOTE: You can change the other properties (color, opacity, etc.) by **right clicking** on the object in the **Navigator** window and selecting **Display**. A pop-up menu will appear with the available properties. You will notice that the muscles are in different colors. Color is not important; it just helps you visualize each structure.

The surface geometry of the muscles and bones are represented as polygonal meshes. The vertices of the polygons are defined with respect to a cartesian coordinate system.

NOTE: You will notice that the muscles are grouped under "Bodies" and not under **Forces → Muscles**. In this model, the muscle structures are not true muscles; they are 3D objects to help you visualize the anatomy. Later in the lab you will create muscle paths from these surfaces.

VII-A.2 Visualize and understand how segments are related by joints.

FIRST Examine the file **Simulation_Lab2_Visualize_Anatomy.osim**.

Locate the where the bodies, joints, generalized coordinates, and kinematic functions are in the file **Simulation_Lab2_Visualize_Anatomy.osim**. It is normal to be overwhelmed when first editing osim files, but once you understand the structure it will be much easier. Ignore the **<defaults>** tag for now; you may collapse it by click on the minus button next to it in Notepad++. You will find all of the bodies defined under the **<BodySet name="">** tag. To easily see this, use the keyboard **shortcut "Alt+5"** to collapse everything to the fifth level. You will be editing a similar osim file

when you build your model so look around and get comfortable with the model file structure.

In this model, the **grnd_pelvis joint** (you will find this under the **<joint>** tag in the **pelvis body**) specifies the transformation from the ground segment to the pelvis segment, and allows the model to be reoriented using the slider bars in the **Coordinates** window in OpenSim. All of the other bodies are connected to the pelvis using a special joint called a weld joint, which is defined in the joint tag to have rotations and translations of zero (i.e., the segment reference frames are coincident). For example, looking at the femur body you will find **<WeldJoint name='pelvis_femur'>** and see that we have not defined any generalized coordinates like in the pelvis body.

SECOND Use the **Property Editor** tool to manipulate the kinematics of a joint.

In OpenSim, you can display the reference frame of the body in the **View** window. **Right click** on the pelvis body in the **Navigator** window and select **"Show Axes."** This shows the origin for the pelvis body, but NOT necessarily the reference frame for a particular joint. To show joint axes, **toggle** the **parent** and **child axes** by **right clicking** on the grnd_pelvis joint. Notice that the pelvis reference frame (child) is coincident with the ground reference frame (parent).

Edit **Simulation_Lab2_Visualize_Anatomy.osim** by going into **Edit → File (.osim,.xml)...** and browse to the model file **Simulation_Lab2_Visualize_Anatomy.osim**. A **Property Editor** will come up; the editor enables you to manipulate the osim file within the OpenSim GUI. Look around at the structure of the model; this is very similar to the osim file you saw previously.

- Navigate to **Model → JointSet → objects → Body (pelvis_femur)**
- Note that the parent body (socket_parent_frame) is the pelvis. **Change** values of **tx, ty, tz** to reposition the femur relative to the pelvis. You can do this by navigating to **"translation"** under **frames → PhysicalOffsetFrame**. Change the values by double-clicking under value and typing in a new value (note: [0], [1], and [2] are translations along X, Y, and Z axes, respectively). Take note that the units are in meters. **Save** the **file** as another name and **reload** it in OpenSim. Did the model changed like your predicted?
- Alternatively, you can use the **Navigator** and **Properties** windows together and find the **"pelvis_offset"** under **Visualize3DAnatomy → Joints → pelvis_femur**.
- Next we are going to **change** values of **rx, ry, rz** to reorient the femur relative to the pelvis. Edit these values in **"orientation"** (or "rotation" in the Navigator window) just below the translations you edited (note: [0], [1], and [2] are rotations about X, Y, and Z axes, respectively). Take note that the units are in radians, not degrees like in the

Coordinates windows. **Save** the **file, reload**, and see the changes you made.

Note: NOT all values can be edited in the GUI and will have to be edited in a text editor.

Each bone is defined relative to its segment reference frame, and the transformation between two segment reference frames is defined by a joint. In the next section of the lab, you will implement a model of the hip joint by redefining the femur reference frame, specifying the translations between the pelvis and the hip center, and describing the rotations between the pelvis and the femur as functions of hip flexion, abduction, and rotation angles.

Close the Visualize 3D Anatomy model when you are done.

VII-B Implement a Kinematic Description of the Hip Joint

Now you are ready to start building your own model!

- Select **File** → **Open Model...** from the OpenSim main menu bar to load the arm model **Simulation_Lab2.osim**
- In addition, open **Simulation_Lab2.osim** in a text or xml editor

Your task in this section is to **implement** a kinematic description of the **hip joint** in the file **Simulation_Lab2.osim**.

Your **hip should**:

- Have **three degrees of freedom** (i.e., ball-and-socket joint)
- **Rotate about** the **center of femoral head** using axes that correspond to flexion/extension, abduction/adduction, and internal/external rotation (i.e., angles that are meaningful to biomechanists and clinicians).

There are several ways to achieve this in OpenSim. One way is outlined below.

VII-B.1 Creating the hip joint in OpenSim.

Implementation of the hip joint kinematics in OpenSim will be greatly simplified if the origin of

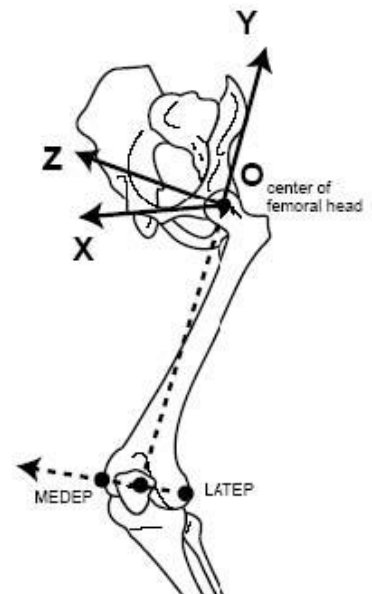


Figure 1. Landmarks that define the femur segment coordinate system. O is the center of the femoral head, MEDEP is the medial epicondyle, and LATEP is the lateral epicondyle.

the femur segment reference frame is located at the desired center of rotation and the axes of the femur segment reference frame are aligned with the desired axes of rotation. In particular, it will be advantageous if the femur reference frame is defined similar to the coordinate system shown in Figure 1.

Please use this coordinate system so that everyone is consistent.

- The **origin** of the desired femur segment reference frame (**Markers** → **hip_center**) is the center of the femoral head.
- The **Y-axis** (superior-inferior) is the vector joining the hip center and the midpoint between the lateral (**Markers** → **LATEP**) and medial (**Markers** → **MEDEP**) epicondyles.
- The **X-axis** (anterior-posterior) is the vector normal to the plane defined by the hip center, the medial epicondyle and the lateral epicondyle.
- The **Z-axis** (medial-lateral) is orthogonal to both the Y and X axes.

FIRST Move the femur joint and body to the correct location.

If you check the **parent** and **child reference frame** of the **pelvis_femur joint** in OpenSim, you will see that it is the same as the origin of the pelvis body. However, we want to **move** this **joint** to the **femoral head** as specified in Figure 1.

- Edit **Simulation_Lab2.osim** using the **Property Editor** or **Navigator-Properties** window combination. Find "**pelvis_offset**" for the **pelvis_femur joint**. This translation is the location of the hip center. Edit "**pelvis_offset**" to **move** it to the **femoral head**. See how your model changed.
- Check your reference frame and make sure that it has been moved to the femoral head location by toggling the **parent** and **child frames** of the **femur**. Your femoral head should be moved away from that reference frame as well. To **move** the **body** to the right location, edit the "**femur_offset**" in the **femur body** to have the femoral head in the correct location. See how your model changed. Check your signs to make sure that it is moving it in the correct position. If you are having trouble, try changing one value at a time so you can see how you are changing the location.
- Is the center of the femoral head at the origin of the femur segment reference frame? Do the axes for the femur body correspond to Figure 1? Where is the femur's reference frame relative to the pelvis? Is the position of the femoral head in the acetabulum?

By now your hip joint should be in the correct location; however, the orientation is not what we want. We want the model to be in the upright standing position when hip flexion, adduction, and rotation are at zero degrees. Note that we have NOT created these coordinates yet since it is still

a weld joint; we will create a custom joint with 6 degrees of freedom (DOF) in the next section. We will determine the required Euler angles using rotation matrices. For more detail on rotation and transformation matrices, refer to Craig (1989).

SECOND: Determine the **3x3 rotation matrix** ${}_{femur}^{base}T$ that describes the desired **femur** reference frame (femur) **relative to** the **current reference frame** (base):

$${}_{femur}^{base}T = [{}_{femur}^{base}R] = [{}_{femur}^{base}X \quad {}_{femur}^{base}Y \quad {}_{femur}^{base}Z]$$

where ${}_{femur}^{base}R$ is a 3x3 rotation matrix that describes the orientation of the desired femur reference frame relative to the current base reference frame. The **3x1 unit vectors** ${}_{femur}^{base}X$, ${}_{femur}^{base}Y$, and ${}_{femur}^{base}Z$ represent the X, Y, and Z axes of the desired femur reference frame relative to the current base reference frame, respectively.

To help you out, we've added the landmarks with markers in the **Simulation_Lab2.osim** model file. You can find the values by using the **Marker Editor (Edit → Markers)** or **right-clicking** the **markers** to view the X, Y, and Z coordinates of the hip center, the medial epicondyle, and the lateral epicondyle with respect to the current base reference frame. **Write these in your report (Table 1)**. Hint: the hip center is ${}_{femur}^{base}P_0$.

Use these coordinates to establish the **rotation matrix**, ${}_{femur}^{base}T$, and **write this 3x3 matrix in your report (Table 2)**. **Make sure that the X, Y, and Z vectors are unit vectors that define a right-handed coordinate system**. You will have to calculate these unit vectors.

THIRD: Determine the **3x3 rotation matrix** ${}_{base}^{femur}T$ that describes the **current** (base) reference frame **relative to** the **desired femur reference frame** (femur).

To establish your new reference frame for the femur, you need to determine the coordinates of the femur bone vertices with respect to the desired frame.

The rotation matrix obtained in the first step (${}_{femur}^{base}T$) would rotate a bone defined in the desired femur frame (femur) to a bone defined in the current base frame (base). However, you want to rotate the existing femur bone (defined in the current base frame) to a bone defined in the desired femur frame. The **rotation matrix** that does this is the **inverse** of ${}_{femur}^{base}T$:

$${}_{base}^{femur}T = ({}_{femur}^{base}T)^{-1}$$

Calculate ${}_{base}^{femur}T$ and **write the matrix in your report (Table 3)**.

FOURTH: Parameterize the rotation matrix by a set of **three rotations** φ , θ , and ϕ :

$${}_{base}^{femur}T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot [Rot(z, \phi)] \cdot [Rot(y, \theta)] \cdot [Rot(x, \varphi)]$$

where $Rot(z, \varphi)$ is a 3x3 rotation matrix that represents a rotation about the Z axis by φ degrees, $Rot(y, \theta)$ represents a rotation about the Y axis by θ degrees, and $Rot(x, \phi)$ represents a rotation about the X axis by ϕ degrees. You are welcome to write your own code that extracts φ , θ , and ϕ from ${}_{base}^{femur}T$ or you may use MATLAB to call the function, **getRPY.m**, which has been provided. The function **getRPY.m** reads a 3x3 rotation matrix and outputs φ , θ , and ϕ . See the code for more details.

Write the values of φ , θ , and ϕ in your report (Table 4). Note that the angles output from the MATLAB function are in degrees, they need to be converted to radians when editing osim files.

FIFTH: Rotate the **femur** to the **correct orientation**

Using the angles you just calculated (φ , θ , and ϕ) in the previous section, you should **rotate** your **femur body** so that it is in the **"scanned" position** at these joint angles. To do this, find the **orientation** property in the Property Editor or the **Rotation** in the Properties window for the pelvis_femur joint in your model and edit this to the appropriate angles. Be sure they correspond to the correct rotations (OpenSim's default rotation sequence is always <x y z>) and make sure it is in radians. See how your model changed. The femur body should be in a standing position. You will be able to check once you have created your joint with hip_flexion, hip_adduction, and hip_rotation.

VII-B.2 Implement the hip kinematics

The **pelvis_femur joint** you just modified in the previous section was a weld joint. A weld joint fixes the location of two bodies; we do not want this for the hip! We want to create a movable joint. In OpenSim, this is known as a **Custom Joint**. Custom joints can be modeled a variety of joints such as pins, slider, universal, etc. Joint kinematics are specified as a series of body-fixed translations and rotations that relate the position and orientation of one body segment to another. The transformations are performed in a user-defined order, and consist of three orthogonal translations and three rotations around user-defined axes. Each translation and rotation may be defined as a constant or as a function of a generalized coordinate.

The **hip joint** in your model will specify the **transformation**, ${}_{femur}^{pelvis}T$, from the **pelvis body** to the **femur body** as a function of the three generalized coordinates hip flexion, hip adduction, and hip rotation (Figure 2).

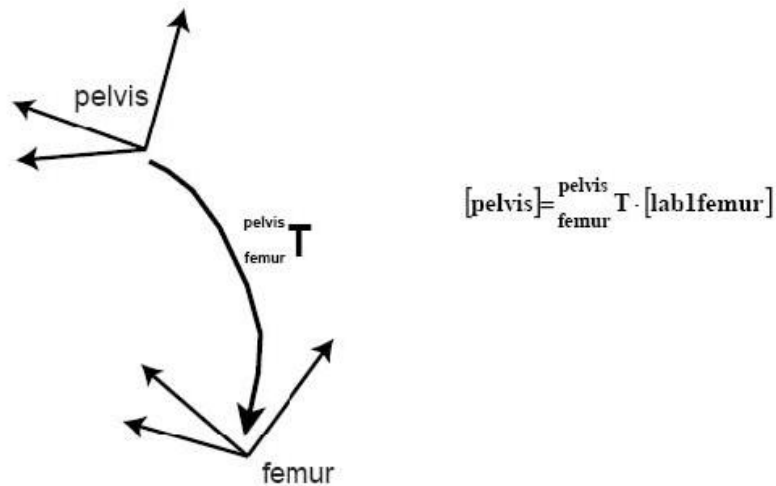


Figure 2. Schematic of the transformation that defines the hip joint

You can change the custom joint in a text or xml editor. Currently OpenSim does NOT allow you to edit this joint in the GUI.

First define coordinate for each translation and rotation. You can create objects under a new tag **<coordinates>**. To define your coordinate, the tag should look something like this **<Coordinate name="name_of_your_coordinate">**. Again, refer to the example **Simulation_Lab2_Arm_Example.osim** model for the correct xml tags for a custom joint as well as the OpenSim User's Guide. Do not worry about the default and initial values for now. The **<range>** tag defines the range of your motion. Note that if it is rotational, this value is in radians and if it is translational, it is in meters. Make sure that the coordinates are not locked or clamped. For example, the following is a **hip flexion coordinate**:

```
<coordinates>
  <Coordinate name="hip_flexion_l">
    <default_value> 0.0 </default_value>
    <default_speed_value> 0.0 </default_speed_value>
    <range> -2.09439510 2.09439510 </range>
    <clamped> false </clamped>
    <locked> false </locked>
    <prescribed_function/>
  </Coordinate>
</coordinates>
```

Next you will define the transform axis for your coordinates you just created. This is under a new tag **<SpatialTransform>**. You can copy this section from one of the custom joints in the **Simulation_Lab2_Arm_Example.osim** model or another OpenSim model and modify it. Below is an example of a **transform axis** for **hip flexion**:

```

<SpatialTransform>
  <TransformAxis name="rotation1">
    <coordinates> hip_flexion_l </coordinates>
    <axis> 0 0 1</axis>
    <LinearFunction name="function">
      <coefficients> 1 0</coefficients>
    </LinearFunction>
  </TransformAxis>
</SpatialTransform>

```

The **<LinearFunction>** tag determines what kind of function the axes will use for the rotation. For our example, it is a linear 1-to-1 function without complex coupling of any kind. The **<coordinates>** tag refers to the coordinates (that you just created previously) that this axis is transforming. The **<axis>** tag defines if the coordinates are changing in the x, y, or z axis (or a combination of the three axes). For this example, hip_flexion_l would rotate about the z axis. There should be rotation1, rotation2, rotation3, translation1, translation2, and translation3 transform axes to define all 6 degrees-of-freedom (DOF). **Define your axes such that hip flexion, adduction, and internal rotation are positive!**

Save your new model as **Simulation_Lab2_with_Hip.osim**.

By now your model should look like it is in the standing position when hip flexion, adduction, and rotation are at zero degrees and have 6 DOF. The femur was placed in the following joint angles during MR imaging:

```

hip_flexion = 24.6 degrees
hip_adduction = 5.45 degrees
hip_rotation = 16.0 degrees

```

Place your femur in these joint angles; they should overlap your original femur. You can overlay the original **Simulation_Lab2.osim** model to check that your femur is in the correct position at the given joint angles. If there is already a model loaded in OpenSim, OpenSim will load the new model with a 2 m offset in the x axis by default. Right click on the new loaded file and select **Display → Model Offset...** to edit the offset, change the X, Y, and Z Offset values to "0" so the models are overlapping. You can change the color and/or opacity of bodies in the various models to differentiate them.

If your femur is not in the correct orientation at the "scanned" joint angles, check your transform axes to make sure you are defining them correctly.

VII-C Implement Kinematic Descriptions of the Knee Joint

Now that your model has a functional hip joint, your next step is to specify the kinematics of the knee. In particular, you will define the motions between

the femur and tibia (the tibiofemoral joint). As you may already know, the tibiofemoral kinematics are more complex than the hip kinematics. To accurately describe the rolling and sliding motion of the tibia relative to the femur, 3D rotations and translations are needed. You might wonder, can't I just represent the knee as a simple hinge joint in the sagittal plane? When is it important to represent the relative translations between the bones (rolling and sliding) and the 3D rotations between the bones (the screw-home mechanism)? Do differences in the knee kinematics really affect the muscle moment arms enough to matter? Let's investigate these issues.

You will implement two descriptions of tibiofemoral kinematics, each with a different degree of complexity:

1. **Knee Joint #1** Hinge joint about an axis through the epicondyles.
2. **Knee Joint #2** Joint with relative translations and 3D rotations between the femur and tibia.

VII-C.1 Create Knee Joint #1.

The tibia and fibula bones that comprise the **tibia body** in **Simulation_Lab2.osim** are defined relative to the base reference frame in the scanned position. The bones **tibia_knee1.vtp** and **fibula_knee1.vtp** has its origin placed so that when it is coincident with the midpoint between the medial and lateral epicondyles, its reference frame is aligned with the reference frame of the femur at full extension (Figure 3). To change the bones, go into in **Display Geometry** under the **<attached_geometry>** and **change** the **vtp file** under **<mesh_file>** for the **tibia** and **fibula**.

Create Knee joint 1 (a hinge joint about an axis through the epicondyles) by modifying the knee joint (similar to the previous part implementing the hip joint) in your newly made **Simulation_Lab2_with_Hip.osim** with the following 3 conditions:

1. Define the **location** to correctly locate the **tibia** reference frame at the **midpoint** between the **medial and lateral epicondyles** (the reference frame origin is coincident with the midpoint)
2. Define the **orientation** to correctly orient the **tibia** reference frame axes to **align** with the **femur** reference frame axes when the knee is **at full extension** (Figure 3)
3. Define the **coordinate** and **transform axis** such that the tibia rotates about its **medial-lateral (z) axis** with knee flexion defined as positive



Figure 3. Reference frames for the femur and tibia bones in full knee extension.

Save your new model as **Simulation_Lab2_with_Hip_Knee1.osim**.

Inspect how the tibia moves about the femur for different angles. Does the tibia have rolling and sliding motion? Does it exhibit any gapping between the tibial plateau and the femoral condyles?

To check your knee joint, overlap the original **Simulation_Lab2.osim** over your model. Your tibia should align with the tibia at the scanned position at the following joint angles:

hip_flexion = 24.6 degrees
hip_adduction = 5.45 degrees
hip_rotation = 16.0 degrees
knee_flexion = 22.0 degrees

VII-C.2 Create Knee Joint #2.

Let's take advantage of published experimental measurements of tibiofemoral kinematics. Fortunately, there are researchers that devote their lives to studying the motions of the tibia and femur during knee flexion. You will base Knee Joint #2 (joint with relative translations and 3D rotations between the femur and tibia) on a reliable set of experimental data reported by Walker *et al.* (1988). Walker *et al.* measured tibiofemoral kinematics in 23 knee specimens. They tracked the motions of landmarks on the femur and tibia with radiographs while flexing and extending the knees by attaching a motor to the quadriceps tendon. The landmarks for each specimen were scaled to an average-sized knee and were used to determine the translations and rotations of the femur relative to the tibia through a range of knee flexion. The translations and rotations, averaged for the 23 specimens, were fit to polynomial equations as a function of knee flexion angle.

The polynomial equations reported by Walker *et al.* (1988) describe the varus rotation (VARUS), internal rotation (INTROT), superior translation (yDIS), and anterior translation (zDIS) of the femur relative to the tibia as functions of knee flexion angle (F). Here we've adopted the notation of Walker *et al.* (1998); please see the paper for details. Walker *et al.*'s reference frames for the femur and tibia are

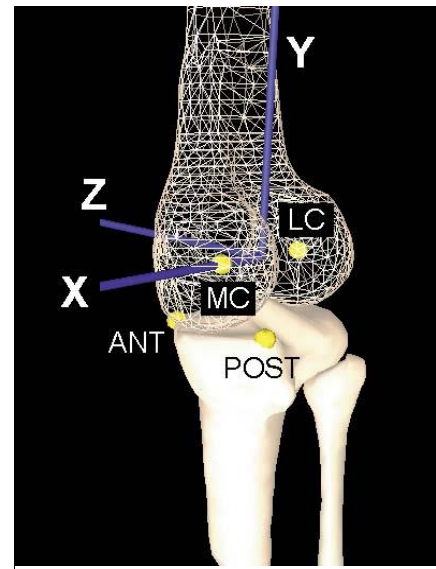


Figure 4. Landmarks that define Walker *et al.*'s femur and tibia segment reference frames for a right limb specimen. Reference frames for the femur and tibia are coincident at full knee extension. MC is the center of a sphere fit to the medial posterior femoral condyle, LC is the center of a sphere fit to the lateral posterior femoral condyle, ANT is an anterior point on the medial tibial plateau, and POST is a posterior point on the medial tibial plateau.

defined as follows when the knee is in full extension (see Figure 4 and below):

- The **origin** is located at the midpoint between centers of spheres fit to the medial femoral condyle (**MC**) and lateral femoral condyle (**LC**).
- The **X axis** (medial-lateral) is the vector that joins MC and LC.
- The **Y axis** (superior - inferior) is the cross product of the vector parallel to the medial tibial plateau (Ant - Post) and the X axis, defined with the knee in full extension.
- The **Z axis** (anterior posterior) is orthogonal to both the Y and X axes.

Your objective in this section is to **implement** Walker *et al.*'s description of **tibiofemoral kinematics** in your lower extremity model. To achieve this, use the polynomial equations in the Walker article to generate the transform axis function for the knee joint. The equations are as follows:

$$tx = (-5.613 \cdot F + 1.108 \cdot F^2 + 7.739) \cdot 0.001$$
$$ty = ((-6.284 \cdot F + 3.183 \cdot F^2 - 0.6314 \cdot F^3) - 2.682) \cdot 0.001$$

NOTE: The equations are slightly different than the Walker article and we added an offset in the z-directions due to slight differences in the bone geometry. "F" is the knee flexion angle in radians.

To implement this in OpenSim, you should **change** your **Transform Axis** function from linear function to a **spline**. The tag is **<SimmSpline name="">** and you do not need to name the function if you don't want to. To define the spline, you should input **"x" and "y" values**. You will need to generate the "x" and "y" values from the above equations. For any x values, there must be an associated y values. You can have as many ordered pairs as you want to define your function. So for example, your **"x" value is "F"** and your **"y" value is "tx"** for the transform axis about the x-axis. You need to define the corresponding coordinate for your "x" value (in this case, "F"). Below is an example (not the Walker values) of a transform axis with **spline**.

```
<TransformAxis name="translation1">
  <coordinates> knee_flexion </coordinates>
  <axis>1 0 0</axis>
  <SimmSpline name="">
    <x> 0.00 0.20 0.30 0.50 </x>
    <y> 0.05 0.02 0.08 0.04 </y>
  </SimmSpline>
</TransformAxis>
```

In this example, the joint will be at **0.05** meters on the positive x axis for knee flexion angle of **0.00** radians. Similarly, the joint will be at **0.02** meters on the positive x axis for knee flexion angle of **0.20** radians.

In the end, you should have **added** a two **additional translation degrees-of-freedom** in your knee joint. You may **use tibia_knee2.vtp** and **fibula_knee2.vtp** instead of s1_tibia.vtp and s1_fibula.vtp. Be very careful to account for differences in reference frames and sign conventions.

Your **kinematic functions** should define **the knee translations** for **knee flexion, F**, over a 0 to 120 degree range of the generalized coordinate. Make sure that the signs of the rotations and translations are consistent with your model.

Save your new model as **Simulation_Lab2_with_Hip_Knee2.osim**.

Now, **reload** your model and check Knee Joint #2.

Like the previous section, your tibia should align with the “scanned” tibia bone when the knee_flexion angle is 22.0 degrees.

Inspect how the tibia moves about the femur. Does the tibia have rolling and sliding motion? Does it exhibit any gapping between the tibial plateau and the femoral condyles? How does the motion of Knee Joint #2 differ from the motion of Knee Joint #1? Does Knee Joint #2 look better than the hinge knee? Briefly **answer these questions in your written report**.

VII-D Specify Muscle-Tendon Paths for the Semitendinosus

You have now completed the most rigorous part of the lab. In this section, you will represent the path geometry of the semitendinosus muscle in OpenSim using a series of line segments. Your goal is to develop and implement a method to locate the muscle attachments and to specify wrapping surfaces and via points as needed to accurately characterize the muscle path over a range of motion. The hip and knee flexion moment arms computed with your model of the specimen should compare well with the muscle moment arms determined experimentally on the same specimen. Your method should be repeatable; that is, based on a written description of your method, someone else should be able to reproduce your muscle path geometry.

You might wonder, can't I just represent the muscle as a straight line from origin to insertion? When is it important to introduce via points and wrapping surfaces to simulate interactions between the muscles and surrounding structures? Do differences in the knee kinematics really affect the muscle moment arms enough to matter? Let's investigate these issues.

You will define three representations of the semitendinosus muscle path:

1. **Muscle Path #1** Straight line from origin to insertion, attached to the tibia in knee #2

2. **Muscle Path #2** Series of straight lines constrained by via points and/or wrapping surfaces, attached to the tibia in knee #2
3. **Muscle Path #3** Same muscle as Muscle Path #2, except attached to the tibia in knee #1

VII-D.1 Create and Evaluate Muscle Path #1.

VII-D.1.1 Represent the semitendinosus as a straight line from origin to insertion.

FIRST Approximate the origin and insertion sites of the semitendinosus.

- **Load** your model with the **knee #2**.
- **Put** the model in the **scanned position**:

hip_flexion = 24.6 degrees
 hip_adduction = 5.45 degrees
 hip_rotation = 16.0 degrees
 knee_flexion = 22.0 degrees

Use the **Marker Editor** tool (**Edit → Marker**) to identify the coordinates of bone vertices that approximate the **attachments** of the **semitendinosus** on the pelvis and tibia. We have approximated the muscle origin (-0.098599, -0.119465, -0.082462) and insertion point (0.102716, -0.492008, -0.115147) when the knee is in the scanned position. Add these markers to verify that it is close to the semitendinosus origin and insertion. **Write these coordinates in your report (Table 5).**

SECOND Define the semitendinosus attachments using the Muscle Editor.

- Open the muscle details by navigating **landmark1 muscle** under **Forces** and viewing it in the **Properties** window (you may also use a text or xml editor)
- Note the type of the muscle. It is usually named after Lisa Schutte, the researcher that developed this muscle in 1993. There are other muscles such as Delp1990Muscle and Thelen2003Muscle.
- **Rename landmark1 to semiten_l_s2.**
- Open the **Muscle Geometry Path** editor by selecting the **"geometrypath"** in the muscle's properties. Replace the **first point** with the **semitendinosus origin** that you found. Make sure that the point is **attached** to the **pelvis**. Note that if there are no muscles defined, you have to use the text editor to add muscles to edit. The GUI Property Editor and Muscle Editor do not allow the addition of muscles.
- Next, replace the **second point** with the **semitendinosus insertion** that you found. Make sure that the point is **attached** to the **tibia** body.

- You should now see a straight red line from your origin to your insertion point. **Save** your model and **open** the osim file to see the changes in the file if you used the text editor instead.

VII-D.1.2 Evaluate your straight-line semitendinosus path

Visually inspect your muscle. Does it look anatomical? Does it penetrate any bones?

An important step in building a model is to evaluate its accuracy. It is critical to understand the limitations of your model. Through the process of testing a model, you can often identify ways to make the model more accurate.

In this section, you will evaluate the accuracy of the semitendinosus moment arms predicted by your model with the simple straight-line path. To do this, you will compare the moment arms calculated with your model to the moment arms determined experimentally. We measured the hip extension and knee flexion moment arms of the semitendinosus muscle (and other muscles) in your specimen. We determined the moment arms using the tendon displacement method (An *et al.*, 1984). This experiment involved measuring the excursions of the muscle over a range of hip flexion and knee flexion angles. Moment arms (ma) were calculated as the partial derivative of muscle-tendon length (δt) with respect to joint angle (θ). That is,

$$ma = \frac{\delta t}{\delta \theta}$$

The hip flexion moment arms were determined with the specimen's hip adduction, hip rotation, and knee flexion angles equal to zero. The knee flexion moment arms were determined with the specimen's hip flexion, hip adduction, and hip rotation angles equal to zero. For more details on the experimental setup, see Arnold *et al.*, 2000.

You will compare the moment arms predicted by your model of the semitendinosus to the experimentally determined moment arms.

Plot the hip flexion and knee flexion moment arms of your semiten_l_s2 muscle.

To **plot** the **hip flexion moment arms** versus hip flexion angle:

- Open the **Plot** tool (**Tools** → **Plot...**)
- **Click Y-Quantity...** select **moment arm** → **hip_flexion** (or whatever you named your hip coordinate)
- **Muscles...** button should be able to be selected now, click on it and select your semiten_l_s2 muscle.
- **Click X-Quantity...** and select your **hip_flexion** coordinate again. This will make a plot of your moment arm of your muscle as a function of hip flexion angle

- **Click** on **Add** button to the right to make your curve. Click on Properties or right click and select properties to adjust your plot if you need to (range, axes, etc.)

NOTE: You can export your plot as an image or export your data into a storage file (.sto) or motion file (.mot). You can open both formats with Excel if you want to modify the data or plot it using Excel. You will need to export your data so you can compare it with the experimental values.

To **plot** the **knee flexion moment arms** versus knee flexion angle, do the same thing except select your knee flexion coordinate.

Plot the **experimental data superimposed on your curves**. The data for the experimental moment arm is available in an Excel file (**Simulation_Lab2_experimental_moment_arm.xlsx**). Plots are already created in the file; you may simply **add your data** to the plots to compare your values. You may format it to fit your data. Make sure you superimpose your data to the appropriate experimental data plot. **Include these plots in your written report, and comment on the validity of representing the semitendinosus path geometry as a straight line from origin to insertion.**

VII-D.2 Create and Evaluate Muscle Path #2.

How well did your straight-line path match the experimental data? Here's your chance to develop an improved method for defining the muscle path over a range of motion. You have **three options** for **refining** the **path geometry** of a muscle in OpenSim:

- Refine or add more attachment points
- Add via points
- Introduce wrapping surfaces

VII-D.2.1 Refine your semitendinosus path by specifying via points and wrapping surfaces.

Copy your **previous model** with muscle #1 so you may modify it. Name your **new muscle semiten_I_2**.

Refine the **attachment sites** for your new muscle. **Add** a **wrapping surface** and any necessary via points (Refer to the OpenSim user guide on how to do this). **Save** your **changes**.

Iterate until you are satisfied with the **agreement** between the **hip** and **knee flexion moment arms** calculated with your model and the moment arms obtained from the experiment. When defining your **muscle path**, you should follow these **rules**:

- Attachments must lie within the 3D muscle surface and on the surface of the pelvis (origin) and tibia (insertion).

- Via points must lie within the 3D muscle surface when the limb is in the scanned position.
- Only one wrapping surface can be used.

The **optimal muscle path** will:

- Not penetrate bones or other muscles over a -15 to 90 degree range of hip flexion and a 0 to 120 degree range of knee flexion
- Predict hip and knee flexion moment arms that compare well with the moment arms determined experimentally (average error less than 10%)
- Predict moment arms curves that are continuous and smooth

Once you have settled on your final semitendinosus muscle path, **plot the hip and knee flexion moment arms compared to the experimental data. Include these plots in your report.** Summarize your method for representing the muscle path geometry. How did you define the attachments?

What was your rationale for locating the wrapping surface and/or via points? Which of these parameters are the moment arms most sensitive to? If you didn't have experimental data for reference, how confident would you be in the moment arms predicted by a kinematic musculoskeletal model?

VII-D.3 Create and Evaluate Muscle Path #3.

In this section, you will study the sensitivity of the semitendinosus moment arms to knee kinematics. Specifically, you will **add a new muscle, semiten_l_hinge**. Semiten_l_hinge should be an **exact copy of your semiten_l_2 muscle, except attached to your hinged knee model**. You will then **compare the knee flexion moment arms of semiten_l_hinge** computed about **Knee Joint #1** (hinge joint about an axis through the epicondyles) **with** the knee flexion moment arms of **semiten_l_2** computed about **Knee Joint #2** (joint with relative translations and 3D rotations between the femur and tibia).

VII-D.3.1 Add Muscle Path #3 to Knee Joint #1 model.

Add a new muscle called semiten_l_hinge to your model with Knee Joint #1.

The easiest way is to **copy** your **semiten_l_2** muscle in your osim model file **from Knee joint #2 to Knee Joint #1** using a text editor. Make sure your body reference(s) are correct.

The other way is to repeat what you did for the previous section to create your muscle on the Knee Joint #1 model using the Muscle Editor. Refer to the previous section if you need help.

VII-D.3.2 Analyze the sensitivity of the semitendinosus moment arms to knee kinematics.

Plot the knee flexion moment arms for the semiten_l_2 and semiten_l_hinge muscles compared to the experimental data. **Include these plots in your report.**

Summarize your knee kinematics-moment arms sensitivity study by answering the following questions:

- How and why are the moment arms of the two muscles different?
- Do you think the differences in the knee kinematics affect the muscle moment arms enough to matter?
- For what applications might a hinge joint knee be sufficient? When would a hinge knee not be sufficient?

VII-E Discussion

In your written report, **comment on the factors that influence the accuracy of muscle moment arms computed with a kinematic musculoskeletal model.**

What techniques might you use (computational or experimental) to increase the accuracy of a model if necessary?

VIII. References

1. An KN, Takahashi K, Harrigan TP, and Chao EY. Determination of muscle orientations and moment arms . *Journal of Biomechanical Engineering*. 106:280-282, 1984.
2. Arnold AS, Salinas S, Asakawa DJ, and Delp SL. Accuracy of muscle moment arms estimated from MRI-based musculoskeletal models of the lower extremity . *Computer Aided Surgery* 5:108-119, 2000.
3. Craig JJ. *Introduction to Robotics: Mechanics and Control*. Reading, MA: Addison-Wesley Publishing Co. 1989.
4. Delp SL, Hess WE, Hungerford DS, and Jones LC. Variation of hip rotation moment arms with hip flexion . *Journal of Biomechanics* 32:493-501, 1999.
5. Delp SL and Loan JP. A graphics-based software system to develop and analyze models of musculoskeletal structures . *Computers in Biology and Medicine* 25:21-34, 1995.
6. Kurosawa H, Walker PS, Abe S, Garg A, and Hunter T. Geometry and motion of the knee for implant and orthotic design . *Journal of Biomechanics* 18:487-499, 1985.
7. Murray WM, Delp SL, and Buchanan TS. Variation of muscle moment arms with elbow and forearm position . *Journal of Biomechanics* 28:513-525, 1995.
8. Walker PS, Rovick JS, and Robertson DD. The effects of knee brace hinge design and placement on joint mechanics . *Journal of Biomechanics* 21:965-974, 1988.