



OpenSim Data Processing Workflow Analysis of Drop Landing

This document is a modified version of the OpenSim tutorials that are available online (<http://simtk-confluence.stanford.edu:8080/display/OpenSim/Introductory+Examples>). This tutorial is based on motion capture data collected at the KU Leuven (Leuven, Belgium).

Data and Example files can be downloaded from <http://simtk-confluence.stanford.edu:8080/display/OpenSim/ESB+2017>.

I. Objectives

The purpose of this tutorial is to demonstrate how OpenSim solves an inverse kinematics and inverse dynamics problem using experimental data. To diagnose movement disorders and study human movement, biomechanists frequently ask human subjects to perform movements in a motion capture laboratory and use computational tools to analyze these movements. A common step in analyzing a movement is to compute the joint angles and joint moments of the subject during movement. Often, also quantities such as muscle activations or joint contact forces are evaluated. OpenSim has tools for computing these quantities:

- *Inverse kinematics* is used to compute joint angles.
- *Inverse dynamics* is used to compute net joint reaction forces and net joint moments.
- *Static Optimization* is used to estimate the muscle activations and corresponding forces that produce these net joint moments.
- *Joint Reaction Analysis* is used to compute the joint contact forces (joint bone-on-bone forces).

Inverse kinematics computes the joint angles for a musculoskeletal model that best reproduce the motion of a subject. *Inverse dynamics* then uses joint angles, angular velocities, and angular accelerations of the model, together with the experimental ground reaction forces and moments, to solve for the net reaction forces and net moments at each of the joints. *Static Optimization* estimates muscle forces from the joint kinematics and external forces and moments using an optimization approach. *Joint Reaction Analysis* uses the results from *Inverse Kinematics* and *Static Optimization* to compute joint contact forces.

Each section of the tutorial guides you in using certain tools within and asks you to answer a few questions. The menu titles and option names you must select and any commands you must type to run OpenSim will appear in **bold face**. The questions can be answered based on information from OpenSim and basic knowledge of the human musculoskeletal system.

II. Generic Musculoskeletal Model

In this tutorial, you will be using a generic musculoskeletal model with 23 degrees of freedom and actuated by 92 muscles entitled *gait2392_Simbody*.

To load the generic musculoskeletal model into OpenSim:

- Click the **File** menu and select **Open Model**.
- Find the **MorningExample/Scaling** folder
- Select the file ***gait2392_simbody_MIFscaled.osim***, and click **Open**.

→ Go to the **Navigator** tab and explore the properties of the model (Bodies, Joints, etc.).

→ Right click on **Bodies**, select **Display** → **Opacity...**

→ Right click on **Forces**, select **Display** → **Hide**

→ Go to the **Coordinates** tab and explore the degrees-of-freedom of each joint

To add a marker set to the model:

- Right click on **Markers** in **Navigator** tab
- Select **Load from file...**
- Find the ***gait2392_Scale_MarkerSet_OSworkshop.xml*** file in the **Scaling** folder (provided by the organizers) and click **Open**.

This marker set includes all markers, which were included in the experimental data collection.

→ Click on the marker names and explore the properties.

Note: You can change the marker names and location either in the GUI or in the .xml file

→ Right click on **Markers**, select **Add New** and add a marker called RHJC (right hip joint center) and RKJC (right knee joint center). Explore the body and location properties of the marker to position the marker at the location of the hip and knee joint center.

Hint: Have a look at the hip and knee joint center markers on the left leg.

→ Save the updated model (**File – Save Model As...**)

Questions

- *How many segments does the model have?*
- *How many degrees-of-freedom do the hip, knee and ankle joint have?*

III. Scaling a Musculoskeletal Model

The purpose of scaling a generic musculoskeletal model is to modify the anthropometry, or physical dimensions, of the generic model so that it matches the anthropometry of a particular subject. Scaling is one of the most important steps in solving inverse kinematics and inverse dynamics problems because these solutions are sensitive to the accuracy of the scaling step. In OpenSim, the scaling step adjusts both the mass properties (mass and inertia tensor), as well as the dimensions of the body segments. Scaling can be performed using a combination of two methods:

- *Measurement-based Scaling:* This type of scaling determines scale factors for a body segment by comparing distance measurements between specified landmarks on the model, known as *virtual markers*, and the corresponding *experimental marker* positions.
- *Manual Scaling:* This type of scaling allows the user to scale a segment based on some predetermined scale factor. Manual scaling is sometimes necessary when suitable marker data are not available, or if the scale factors were determined using an alternative algorithm.

To scale the generic model:

Click the **Tools** menu and select **Scale Model**.

At the bottom of the *Scale Tool* dialog, click **Load** to input a settings file.

In the file browser select the file ***OSworkshop_ScaleSetup.xml*** in the **Scaling** folder (provided by the organizers) and click **Open**.

This *xml* file contains pre-configured settings to scale the generic musculoskeletal model to the dimensions of the subject. Notice all of the textboxes in the dialog were filled in appropriately.

→ Click on all the tabs in the Scale Tool window and explore the settings

Questions

- *Based on information in the Scale Tool dialog, what is the mass of the generic musculoskeletal model? What was the mass of the subject?*

- Which body segment was scaled manually?
- Based on information in the Inverse Kinematics Tool dialog, at what frequency was the experimental motion data captured? Hint: Look for the box titled Marker Data.
- The thigh segment is often scaled using the distance between the ASIS and lateral knee marker but can be also scaled using the distance between the hip and knee joint centers. What is the benefit of using joint centers for scaling and in which population might this be useful?

To complete the scale step:

- In the *Scale Tool* dialog, click **Run**. Then click **Close**.
- To save the scaled model, click **File** and select **Save Model**.

IV. Inverse Kinematics

Kinematics is the study of motion without considering the forces and moments that produce that motion. Thus, to perform kinematical analyses, such as inverse kinematics, mass and inertia properties are not needed. The purpose of inverse kinematics is to find the joint angles of the model that best reproduce the experimental kinematics of a particular subject. In this tutorial, the experimental kinematics used by the inverse kinematics tool are based on *experimental marker* positions.

The inverse kinematics tool goes through each time step, or frame, of recorded motion and computes the set of joint angles that put the model in a configuration that "best matches" the experimental kinematics. OpenSim determines this "best match" by solving a weighted least squares optimization problem with the goal of minimizing *marker error*. For more information about optimization and least-squares problems, see Chapter 1 of *Convex Optimization* by Stephen Boyd and Lieven Vandenberghe (<http://www.stanford.edu/~boyd/cvxbook/>).

Marker error is defined as the distance between an *experimental marker* and the corresponding *virtual marker*. Each marker has an associated weighting value, specifying how strongly that marker's error term should be minimized in the least squares problem. In each frame, the inverse kinematics tool solves for a vector of generalized coordinates (e.g. joint angles), \mathbf{q} , that minimizes the weighted sum of marker errors, which is expressed as

$$\min_{\mathbf{q}} \left[\sum_{i \in \text{markers}} w_i \|\mathbf{x}_i^{\text{exp}} - \mathbf{x}_i(\mathbf{q})\|^2 \right]$$

where \mathbf{q} is the vector of generalized coordinates (e.g. joint angles), $\mathbf{x}_i^{\text{exp}}$ is the position of *experimental marker* i , $\mathbf{x}_i(\mathbf{q})$ is the position of the corresponding *virtual marker* i (which depends on \mathbf{q}), and w_i is the weight associated with marker i .

To solve the inverse kinematics problem:

- Lock the mtp joints in the **Coordinates** tab by clicking on the 'lock symbol' next to mtp_angle_l and mtp_angle_r coordinates.
- Save the model
- Click the **Tools** menu and select **Inverse Kinematics**.
- In the *Inverse Kinematics Tool* dialog, click **Load** to load an Inverse Kinematics setup file.
- In the file browser select the file *IKsetup.xml* in the **IK** folder (provided by the organizers) and click **Open**.

This *xml* file contains pre-configured settings to solve the inverse kinematics problem for the scaled model. Notice all of the textboxes in the dialog were filled in appropriately.

Questions

- Click the **Weights** tab and scroll through the list of markers in the top half of the dialog. Which markers have weighting values less than 10? Why?

To complete inverse kinematics step:

- In the *Inverse Kinematics Tool*, click **Run**. Then click **Close**.

Note: Even though you closed the dialog, the Inverse Kinematics tool is still running. Notice the progress bar in the lower right-hand corner of the program. Wait until the bar disappears before proceeding.

The model will begin to move slowly, as the inverse kinematics problem is being solved for each frame of the experimental data.

To compare experimental marker data with inverse kinematics results, in the *Navigator* panel, go to **Motions** and right-click on **Results** (this is the output generated by the Inverse Kinematics Tool). Then choose **Associate Motion Data...** from the drop down menu. Choose *Workshop_Tr10_cro.trc* and click **Open**. Model markers or *virtual markers* are shown in pink and *experimental markers* are shown in blue. Hit play in the Motion Toolbar. The *virtual markers* should correspond closely to the *experimental marker* locations as the animation proceeds.

*Note: If using a Virtual Machine on a Mac, Command + Ctrl + Left Click on each motion.

When completed, examine the accuracy of the inverse kinematics solution:

- Click the **Window** menu and select **Messages**.
- The *Messages* window records details of the all steps you have performed. Take a minute to explore the Messages window. Then, **scroll** to the very bottom.
- The next to last line provides the markers errors and model coordinate errors (e.g. joint angle errors) associated with the last frame of the motion.

Note: All marker errors have units in meters, and all coordinate errors have units in radians.

Questions

- *Based on information in the Messages window, what is the root-mean-squared (RMS) error of all the markers in the last frame of the motion? Include units. Does this seem reasonable? Explain.*
- *What was the value of the maximum marker error in the last frame? Include units. Which marker had this maximum error, and why?
Hint: Think about the weighted least squares problem.*

To visualize the inverse kinematics solution, animate the model by using the *motion slider* and *video controls*.

*Remember: You can **loop**  and control the speed of the animation.*

The inverse kinematics solution is saved to *Workshop_Tr10_cro.mot* as specified in the setup file.

Note: Be sure to use the exact file name given, as this file is used to complete the next step.

→ Run inverse kinematics (IK) again and include the knee markers (RKNE, RKNEmed, LKNE, LKNEmed) in the calculations. Don't forget to change the output name before running IK to avoid overwriting the previous output file. Compare the RMS and max marker errors from this and the previous IK solution.

→ Lock the subtalar angle joints and run inverse kinematics (IK) again. Don't forget to change the output name before running IK to avoid overwriting the previous output file. Compare the RMS and max marker errors from this and the first IK solution.

Hint: Compare the experimental marker data with the inverse kinematics results.

→ Run IK again using only some markers at the foot segment (e.g. only the markers included in the Plug-in-Gait marker protocol, i.e. heel and toe markers, etc). Don't forget to change the output name before running IK to avoid overwriting the previous output file. Compare your ankle and subtalar angles with the results from the first IK analysis using the **Plot** tool as described below.

- Click **Tools** and select **Plot**.
- In the *Plotter* window, click the **Y-Quantity** button and select **Load File**.
- In the file browser, go to the *Output* folder, select the **Workshop_Tr_10_cro.mot** file, and click **Open**.
- In the menu, select **ankle_angle_I** and **subtalar_angle_I** by clicking the corresponding checkboxes, then click **OK**.
Note: To find these quantities quickly, type ankle into the pattern text box.
- Click the **X-Quantity** button, select **time**, and click **OK**.
- Back in the *Plotter* window, click **Add** to add the moment curves to the plot.
- Do the previous steps again but use your results from the IK analysis which only included 2 foot markers and compare the obtained kinematic waveforms.

- Print your plot by **right clicking** on the plot and selecting **Print**.
*Note: You can export the plot as an image by **right-clicking** the plot and selecting **Export Image**.*
- After printing the plot and answering the following questions, **close** the *Plotter* window.

Questions

- *Why did including the knee markers increase the marker tracking errors?*
- *Why did locking the subtalar joints increase the marker tracking errors?*
- *If you consider a single degree-of-freedom between the shank and foot segment (locked subtalar joint), how many markers would you need at the foot segment to estimate this motion along this degree-of-freedom?*
- *How many markers would you need at the foot segment if you open the subtalar joint?*

V. Inverse Dynamics

Dynamics is the study of motion *and* the forces and moments that produce that motion. Thus, to perform dynamical analyses, such as inverse dynamics, estimation of mass and inertia is required. The purpose of inverse dynamics is to estimate the forces and moments that cause a particular motion, and its results can be used to infer how muscles are utilized for that motion. To determine these forces and moments, equations of motion for the system are solved iteratively [3]. The equations of motion are derived using the kinematic description and mass properties of a musculoskeletal model. Then, using the joint angles from inverse kinematics and experimental ground reaction force data, the net reaction forces and net moments at each of the joints are calculated such that the dynamic equilibrium conditions and boundary conditions are satisfied [3].

Note: Joint reaction, or inter-segmental, force is the total force acting across a particular joint in a model. This should not be confused with joint bone-on-bone force, which is the actual force seen across the articulating surfaces of the joint and include the effect of muscle activity. For a thorough discussion on this topic see pp 77-79 in [4].

To solve the inverse dynamics problem:

- Click the **Tools** menu and select **Inverse Dynamics**.
- In the *Inverse Dynamics Tool* dialog, click **Load** to load an Inverse Dynamics setup file.
- In the file browser select the file **IDsetup.xml** in the **ID** folder (provided by the organizers) and click **Open**.

*Note: If the **Motion From File** textbox appears red, this means the textbox was filled with an inappropriate file name. Make sure the motion file was saved with the correct file name in the Inverse Kinematics section.*

- Note the folder listed in the **Directory** textbox, located in the **Output** section of the dialog. The storage file containing the inverse dynamics results will be saved in this folder.
- Click **Run** at the bottom of the dialog. Then click **Close**.

When completed, examine the results of the inverse dynamics solution by plotting the net moments at the left and right ankles:

- Click **Tools** and select **Plot**.
- In the *Plotter* window, click the **Y-Quantity** button and select **Load File**.
- In the file browser, go to the *Output* folder, select the file *inverse_dynamics_subTalarUnlocked.sto*, and click **Open**.
- In the menu, select **ankle_angle_r_moment** and **ankle_angle_l_moment** by clicking the corresponding checkboxes, then click **OK**.
Note: To find these quantities quickly, type ankle into the pattern text box.
- Click the **X-Quantity** button, select **time**, and click **OK**.
- Back in the *Plotter* window, click **Add** to add the moment curves to the plot.
- Print your plot by **right clicking** on the plot and selecting **Print**.
*Note: To export the plot as an image by right-clicking the plot and selecting **Export Image**.*
- After printing the plot and answering the following questions, **close** the *Plotter* window.

Questions

- *On your plot of the ankle moments, identify the time interval when the participant was falling, i.e. did not have contact with the floor or box.*
- *Identify when the participant landed on the floor.*
- *Plot your residual forces (*pelvis_tx_force*, *pelvis_ty_force*, *pelvis_tz_force*). Are these forces acceptable? What could have caused these residual forces?*

→ Run Inverse Dynamics again using the model with the locked subtalar joint. Compare the ankle moments with the previous results.

Questions

- *How big are the differences in angle moments? Do the results make sense?*

VI. Static optimization

Static Optimization is a method for estimating muscle activations and muscle forces that satisfy the positions, velocities, accelerations and external forces (e.g., ground reaction forces) of a motion. The technique is called "static" since calculations are performed at each time frame, without integrating the equations of motion between time-steps. Because there is no integration, Static Optimization can be very fast and efficient, but it does ignore activation

dynamics and tendon compliance. (See [Hicks et al., \(2015\)](#) for more details regarding this and similar modeling and simulation choices and their pros and cons.)

As with any analysis or simulation, the quality of the Static Optimization results largely depends on the inputs; the model, motion, and forces. The model should have mass, anthropometry, and strength that represent the experimental participant; while the degrees of freedom and muscle geometry should be appropriate for the questions being asked (e.g., studying an upper body motion would require a more detailed upper extremity muscle set and skeletal geometry). The motion should contain smooth, realistic accelerations and all external forces during the motion (e.g., ground reaction forces) should be accurately measured and applied to the model. Additional forces (i.e., reserves and residuals) are often needed, but should be small enough to not confound analysis. Any issues with your inputs will give you poor results or cause the tool to fail.

To solve the static optimization problem:

- Click the **Tools** menu and select **Static Optimization**.
- In the *Static Optimization Tool* dialog, click **Load** to load a Static Optimization setup file.
- In the file browser select the file *SOsetup.xml* (provided by the organizers) and click **Open**.
- Make sure that the motion that is load in the input is the inverse kinematics solution with the unlocked subtalar joint (*Workshop_Tr10_cro.mot*).
- Click **Run**. Then click **Close**.

To see the forces that were calculated by the optimization:

- Click the **Tools** menu and select **Plot**.
- Click **Y-Quantity** and **Load file**. Open the file containing the calculated forces just created by Static Optimization *OSworkshop_scaled_StaticOptimization_force.sto*.
- Find the **tib_post_I** muscle and click **OK**.
- Click **X-Quantity**, select **time** and click **OK**.

Questions

- *What is the maximal force the tibialis posterior is delivering and at what time in the motion cycle? Does this correspond with the inverse dynamics results?*
- *Also plot the activation of the tibialis posterior. What is the maximal activation this muscle is delivering?*

We added reserve and residual actuators to the model. Reserve actuators can add extra actuation during portions of the gait cycle where muscles are not able to generate sufficient accelerations (e.g., during a spike in acceleration). Residual actuators are "hand of God" forces that account for (what should be small) discrepancies between the model, measured motions, and forces, ensuring that $F=ma$ throughout the analysis. For more details, see section 3.1.3 of Hicks et al., (2015).

To see how much force or torque the reserve and residual actuators are generating you can plot them:

- Click the **Tools** menu and select **Plot**.
- Click **Y-Quantity** and **Load file**. Open the file containing the calculated forces just created by Static Optimization *OSworkshop_scaled_StaticOptimization_force.sto*.
- Find the **ankle_angle_I_reserve** and click **OK**.
- Click **X-Quantity**, select **time** and click **OK**.

Questions

- *How much reserve torque is delivered for the left ankle angle? Do you think this is acceptable?*
- *Also plot the residual actuator FY. What does this actuator represent and do you think the amount of force it is delivering is acceptable?*

Note: The maximal isometric force of all muscles in the model you are using was doubled compared to the original Gait2392 model. When not increasing the maximal isometric forces, several muscles reach their maximal activation level and reserve toques are high. Although the generic model is sufficiently strong to perform less explosive movements, increased maximal isometric forces have been used previously to simulate for example running.

→ Lock the subtalar angle joints and run the static optimization again with the inverse kinematics solution with the locked subtalar joint. Compare the force delivered by the tibialis posterior with and without locking the subtalar joint.

Questions

- *What is the effect of locking the subtalar joint on the force delivered by the tibialis posterior? Explain this effect.*
- *Which other muscle forces do change by locking the subtalar joint? Do forces from muscles that do not actuate the subtalar joint also change?*
- *Make a plot of the moment arm lengths of all muscles actuating the subtalar joint versus the motion. To do so you first need to load the motion for the model. In the plot tool, you can choose for Y-Quantity -> moment arm -> subtalar_angle_I. For the muscles you choose all muscles that actuate the subtalar joint. For X-Quantity you can choose the motion you just loaded for the model. How do these moment arm lengths influence the muscle forces you just calculated?*

VII. Joint Reaction and Muscle Force Direction Analyses

JointReaction is an OpenSim Analysis for calculating resultant forces and moments at joint. Specifically, it calculates the joint forces and moments transferred between consecutive bodies as a result of all loads acting on the model. These forces and moments correspond to the internal loads carried by the joint structure. These loads represent the contributions of all un-modeled joint structures that would produce the desired joint kinematics, such as cartilage contact and any omitted ligaments. The reaction load acts at the joint center (mobilizer frame) of both the parent and child bodies. The loads can be reported and expressed in either the child, parent, or ground frames. The default behavior is to express the force on the child in the ground frame. For more detailed description of the method implementation see *Tibiofemoral Contact Force during Crouch Gait* (Steele et al, 2012).

To run the joint reaction analysis:

- Click the **Tools** menu and select **Analyze**.
- In the *Analyze Tool* dialog, click **Load** to load an Analysis setup file.
- In the file browser select the file *JRLsetup.xml* (provided by the organizers) and click **Open**.
- Make sure to use the inverse kinematics solution obtained with the unlocked subtalar joint (*Workshop_Tr10_cro.mot*).
- Go to the Analyses tab, click on *JointReaction* at the top and click **Edit** at the bottom.
- In this window, you can edit the properties of the Joint Reaction Analysis tool. You can select the bodies for which you want to compute the applied forces (*apply_on_bodies*) as well as the reference frame in which you want the forces to be expressed (*express_in_frame*). You can change both properties by double clicking on their names.
- Make sure that in the *forces_file* field you add the file containing the forces calculated by static optimization (*OSworkshop_scaled_StaticOptimization_force.sto*). You can change this file by double clicking on the path.
- Click **OK**.
- Click **Run**. Then click **Close**.

Questions

- *Plot the joint contact forces at the ankle joint. In which direction are the contact forces highest?*

→ Lock the subtalar angle joints and run the joint reaction analysis again with the inverse kinematics solution and static optimization forces obtained with a locked subtalar joint. Compare the contact forces at the ankle joint with and without locking the subtalar joint.

Questions

- *What is the effect of locking the subtalar joint on the contact forces in the ankle? Why?*
- *Are contact forces in the other joints also affected?*

On simtk.org a plugin is available that computes the direction and application points of the muscle forces. You can download the plugin from https://simtk.org/projects/force_direction.

For more details on the plugin you can read the documentation provided with this simtk project.

→ Download the plugin and install it following the instructions in the documentation.

To run the MuscleForceDirection plugin:

- Click the **Tools** menu and select **Analyze**.
- In the *Analyze Tool* dialog, click **Load** to load an Analysis setup file.
- In the file browser select the file **MFdirection_setup.xml** (provided by the organizers) and click **Open**.
- Make sure to load the inverse kinematics solution obtained with the subtalar joint unlocked (*Workshop_Tr10_cro.mot*).
- Click **Run**. Then click **Close**.

Questions

- *Two files are created, `_MuscleForceDirection_attachments.sto` and `_MuscleForceDirection_vectors.sto`. Which information is stored in these two files?*
- *From the `_MuscleForceDirection_attachments.sto` file, plot the attachments of the `vas_lat_l_X2_on_tibia_l` and `vas_lat_l_X1_on_femur_l` versus time. What is the difference between these two attachment points? Why is `vas_lat_l_X2_on_tibia_l` changing over time, while `vas_lat_l_X1_on_femur_l` is constant for the entire motion? (Hint: check the documentation for the description of the “Anatomical and effective muscle attachments”).*

Credits

Stanford’s OpenSim team

Friedl De Groote, Ilse Jonkers, Mariska Wesseling, Hans Kainz, Sam Van Rossom