Scaling and Inverse Kinematics[[1]](#footnote-1)

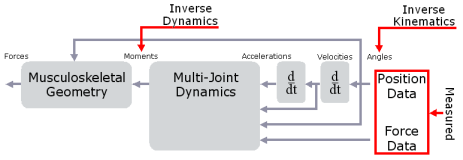
# I. Objectives

## Purpose

## The purpose of this tutorial is to demonstrate how OpenSim solves an inverse kinematics 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 of the subject during movement. OpenSim has tools for computing these quantities:

## *Inverse kinematics* is used to compute joint angles.

## *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. The schematic below shows an overview of the inverse kinematics and inverse dynamics problems.



In this tutorial, you will:

* Become familiar with OpenSim's Scale and Inverse Kinematics tools
* Solve an inverse kinematics problem using experimental data

## Format

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. After you complete the tutorial, feel free to explore OpenSim and the other analysis tools further on your own. Depending on the amount of exploration you do, this tutorial should take 1-2 hours to complete.

# II. Generic Musculoskeletal Model

In this tutorial, you will be using a generic musculoskeletal model with 23 degrees of freedom and actuated by 54 muscles entitled *3DGaitModel2354*. It is a simplified version of the lower-extremity model of Delp et al. [1], modified to include a torso and back joint based on the model of Anderson and Pandy [2].

To load the generic musculoskeletal model into OpenSim:

* Click the **File** menu and select **Open Model**.
* Find the ***Models*** folder, which is located under your OpenSim installation directory, e.g., *C:\OpenSim3.2. (The actual directory name depends on version used and can be changed during installation).*
* Open the ***Gait2354\_Simbody*** folder, select the file ***gait2354\_simbody.osim***, and click **Open**.

The experimental gait data were collected by Jill Higginson and Chand John in the Neuromuscular Biomechanics Lab at the University of Delaware as part of the study cited below. The data include marker trajectories and ground reaction forces for an adult male walking at a self-selected speed on an instrumented split-belt treadmill. Please note that the data distributed with OpenSim is from a different subject than the one described in the paper. Data collection protocols were the same for both subjects.

Chand T. John, Frank C. Anderson, Jill S. Higginson & Scott L. Delp (2012): Stabilisation of walking by intrinsic muscle properties revealed in a three-dimensional muscle-driven simulation, Computer Methods in Biomechanics and Biomedical Engineering.

# 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:

(1) 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.

(2) 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, ensure that you are in the **Gait2354\_Simbody** folder, select the file **subject01\_Setup\_Scale.xml** 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.

## Questions

1. Based on information in the Scale Tool dialog, what is the mass of the generic musculoskeletal model? What was the mass of the subject?
2. To see the loaded scale factors, click on the **Scale Factors** tab. Which body segments were scaled manually?
3. 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.

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.**
* Ensure that you are in the **Gait2354\_Simbody** folder, type **gait2354\_scaled.osim** into the **File name** textbox, and click **Save**.

When complete, a new, scaled model entitled subject01 will appear in View window. Notice the blue spheres around the new model. These blue spheres graphically represent the experimental markers from the motion capture data used in the measurement-based scaling.

At this point, you may close the generic model (right-click the model name in the Navigator window, and select "close") or hide the model (right-click the model name, and select Display >> Hide).

# 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 match" 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), **q**, that minimizes the weighted sum of marker errors, which is expressed as,

http://simtk-confluence.stanford.edu:8080/download/attachments/5115855/Ik.png?version=2&modificationDate=1379004866042

where **q** is the vector of generalized coordinates (e.g., joint angles), **x**i**exp** is the position of experimental marker i, **x**i(**q**) is the position of the corresponding virtual marker i (which depends on **q**), and wi is the weight associated with marker i.

To solve the inverse kinematics problem:

* 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, ensure that you are in the **Gait2354\_Simbody** folder, select the file **subject01\_Setup\_IK.xml** 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

1. 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 one? Why?* *Hint: Think about joints that have not been modeled*.

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** (which are what the Inverse Kinematics Tool just generated). Then choose **Associate Motion Data*...*** from the drop down menu. Choose *subject\_01\_walk1.trc*and click **Open**. Model markers are *virtual markers* 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*.

1. 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.
2. 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. The model should walk through one full gait cycle. Remember: You can***loop****http://simtk-confluence.stanford.edu:8080/download/attachments/5115855/worddav613169f3722e482985e2b7ba46883866.png?version=1&modificationDate=1379004377736*and control the speed of the animation.

The inverse kinematics solution is saved to subject01\_walk1\_ik.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.

# V. Understanding of Joint Angles

## Questions

1. *To check our understanding of where these joint angles come from, find the flexion, adduction, and internal rotation joint angles from the following rotation matrix. Be sure to show all work to receive full credit.*

PelvisRFemur =

# References

1. Delp, S.L., Loan, J.P., Hoy, M.G., Zajac, F.E., Topp E.L., Rosen, J.M. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. IEEE Transactions on Biomedical Engineering, vol. 37, pp. 757-767, 1990.
2. Anderson, F.C., Pandy, M.G. A dynamic optimization solution for vertical jumping in three dimensions. Computer Methods in Biomechanical and Biomedical Engineering, vol. 2, pp. 201-231, 1999.
3. Kuo, A.D. A least squares estimation approach to improving the precision of inverse dynamics computations, Journal of Biomechanical Engineering, vol. 120, pp. 148-159, 1998.
4. Winter, D.A. Biomechanics and Motor Control of Human Movement, Wiley and Sons, pp. 77-79, 1990.
5. Thelen, D.G., Anderson, F.C. Using computed muscle control to generate forward dynamic simulations of human walking from experimental data, Journal of Biomechanics, vol. 39, pp. 1107-1115, 2006.
6. John, C.T., Anderson, F.C., Guendelman, E., Arnold, A.S., Delp, S.L. An algorithm for generating muscle-actuated simulations of long-duration movements, Biomedical Computation at Stanford (BCATS) Symposium, Stanford University, 21 October 2006, Poster Presentation.
7. Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., Thelen, D.G. OpenSim: Open-source software to create and analyze dynamic simulations of movement. IEEE Transactions on Biomedical Engineering, vol. 55, pp. 1940-1950, 2007.

# Deliverables

Answer all questions posed in the tutorial and turn in your report electronically (as a .docx format) using Blackboard. Restate each question, followed by your answer. Be sure to include plots and/or figures to support your answers. For example, if you answer a question with ‘the knee flexion moment arm of the hamstrings decreased with knee flexion angle’, be sure to include a plot to support your statement. The report will be graded as follows:

|  |  |
| --- | --- |
| **Question** | **Points Possible** |
| 1. Based on information in the Scale Tool dialog, what is the mass of the generic musculoskeletal model (1pt)? What was the mass of the subject (1pt)? | 2 |
| 2. Which body segments were scaled manually (1pt)? | 1 |
| 3. Based on information in the Inverse Kinematics Tooldialog, at what frequency was the experimental motion data captured (1pt)? | 1 |
| 4. Which markers have weighting values less than one (1pt)? Why (1pt)? | 2 |
| 5. 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 (1pt)? Include units. Does this seem reasonable (1pt)? Explain. | 2 |
| 6. What was the value of the maximum marker error in the last frame (1pt)? Include units. Which marker had this maximum error (1pt), and why (1pt)? | 3 |
| 7. Find the flexion (1pt), adduction (1pt), and internal rotation (1pt) joint angles from the rotation matrix. | 3 |
| **Total** | **14** |

1. This lab adapted from <http://simtk-confluence.stanford.edu:8080/display/OpenSim/Tutorial+3+-+Scaling%2C+Inverse+Kinematics%2C+and+Inverse+Dynamics> [↑](#footnote-ref-1)