Release 0.6

June 1, 2006

Gait Workflow User's Guide

The National Center for

Physics-Based Simulation of

Biological Structures

at Stanford

Acknowledgements

OpenSim is a part of SimTK and the Simbios project funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 GM072970. Information on the National Centers for Biomedical Computing can be at http://nihroadmap.nih.gov/bioinformatics.

Authors

Frank C. Anderson, Ph.D.

Ayman Habib, Ph.D.

Peter Loan, M.S.

Scott L. Delp, Ph.D.

Intended Audience

This user's guide is intended for users of OpenSim. OpenSim is an object-oriented modeling framework written in C++ for the simulation, control, and analysis of neuromusculoskeletal systems. User's of OpenSim are likely to range from programmer's and engineers who model the neuromusculoskeletal system to scientists and clinicians who use simulation to investigate the biomechanics of movement. This manual, because it largely details the software engineering aspects of OpenSim, is likely to be of greater interest to programmers and engineers. Although not required, a working knowledge of SIMM from Musculographics, Inc. and familiarity with the C++ programming language is recommended.

Trademarks & Copyright

Notes

OpenSim 0.6 is an alpha release not yet intended for general public use. As such, the documentation and source code for OpenSim is undergoing heavy revision and bug fixing.

During the coming months, a series of releases will be rolled out (0.7, 0.8, and 0.9), leading up to a full public release with OpenSim1.0. Until Release 1.0, access to OpenSim source code, libraries, and executables will be restricted to OpenSim project members. As of Release 0.6, there are a total of 20 project member. A substantial number of these members are from institutions outside Stanford University, including the University of Wisconsin-Madison, the University of Texas at Austin, the National Institutes of Health, and the Royal Veterinary College of the University of London. The number of OpenSim project members is expected to grow, and the OpenSim development team will be actively seeking experts in biomechanical simulation to become new members to help test, augment, and refine OpenSim.

While OpenSim source code and downloads are currently restricted to project members, the documentation for OpenSim has been made accessible to the general public. We do this to generate interest in OpenSim, document the growing capabilities of OpenSim, and solicit feedback. Those interested in OpenSim are encouraged to download the documentation and send questions and feedback to the project administrators, Frank C. Anderson and Ayman Habib. The project administrators can be contacted through the OpenSim project on SimTK.org.

If you would like to joint the development team, you may direct inquiries also to the project administrators (see above). Keep in mind that, especially during the early releases of OpenSim, the number of members will be kept small in order to make the testing process more manageable. As OpenSim becomes a more hardened and robust framework, the number of project members will be allowed to grow more rapidly.

Feedback on this User's Guide is welcomed!

Table of Contents

Chapter 1

What is the Gait Workflow?

The Gait Workflow is a set of tools, written on top of OpenSim, for generating subject-specific, muscle-actuated simulations of gait. (For information about OpenSim, see the OpenSim Developer's Guide.) The workflow takes as input a generic musculoskeletal model and experimental gait data (kinematics and ground reaction forces) and consists of 4 main steps, each of which is carried out by a command-line executable (Fig. 1.1).

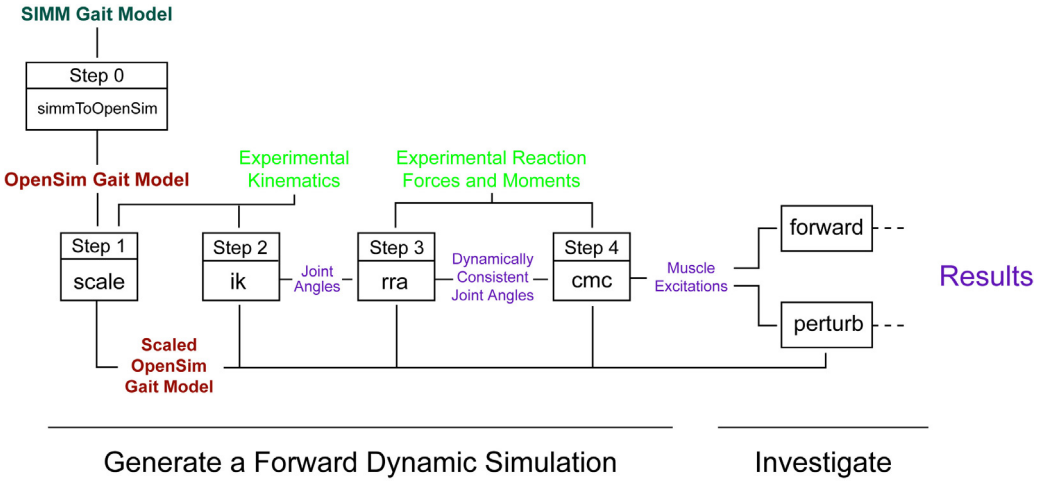

Figure 1.1: Schematic of the Gait Workflow.

If necessary, before beginning the workflow, a SIMM gait model (described by a SIMM joint and muscle file) is converted to an OpenSim gait model by running `simmToOpenSim.exe` (Step 0). In Step 1, by running `scale.exe`, the generic gait model is scaled to a particular subject. In Step 2, by running `ik.exe`, an inverse-kinematics problem is solved to find the set of joint angles that best reproduce the input kinematics. In Step 3, by running `rra.exe`, a residual reduction algorithm is applied to make small changes in the joint angles so they are more dynamically consistent with the ground reaction forces. Finally, in Step 4, by running `cmc.exe`, Computed Muscle Control (CMC) is used to solve for muscle excitations that drive the gait model to track the processed kinematics. Once a forward dynamic simulation is generated, additional executables can be used to investigate the simulation. The executable `forward.exe` can be used to run

analyses on the generated simulation. The executable `perturb.exe` can be used to run perturbations for quantifying muscle function.

Prerequisites

To run the Gait Workflow, a user must have the following:

1. An installation of OpenSim. The OpenSim Developer's Guide contains instructions for installing OpenSim.

2. If you would like to visualize output at various stages of execution in the Workflow, an installation of SIMM from Musculographics is needed.

3. A text editor. An editor specifically designed for editing XML (Extensible Markup Language) may be helpful, as many of the setup files for the Workflow are in XML format.

OpenSim augments the functionality of SIMM and the SIMM Dynamics Pipeline by providing advanced simulation and control capabilities. In addition, the object-oriented, modular design of OpenSim allows users to extend its functionality and share functionality with other OpenSim users.

Workflow Example

The downloaded distribution of OpenSim contains a set of sample data files that can be used to run the Gait Workflow. The data files are located in `OpenSim/Documentaton/WorkflowExample` at the install location for OpenSim, which on Windows systems is usually `C:\Program Files`. The data files are also available by checking out the OpenSim source code repository; they are located in `/Trunk/Documentation/WorkflowExample` of the repository.

Table of Files

Following are a series of schematics that show how the Workflow is executed (Figs. 1.2 – 1.5). Each of the schematics list the possible input and output files for each of the main steps in the Workflow and also contains a list of user actions for the user to follow to execute the Workflow.

For more details on each of the workflow steps, consult Chapters 2 – 5.

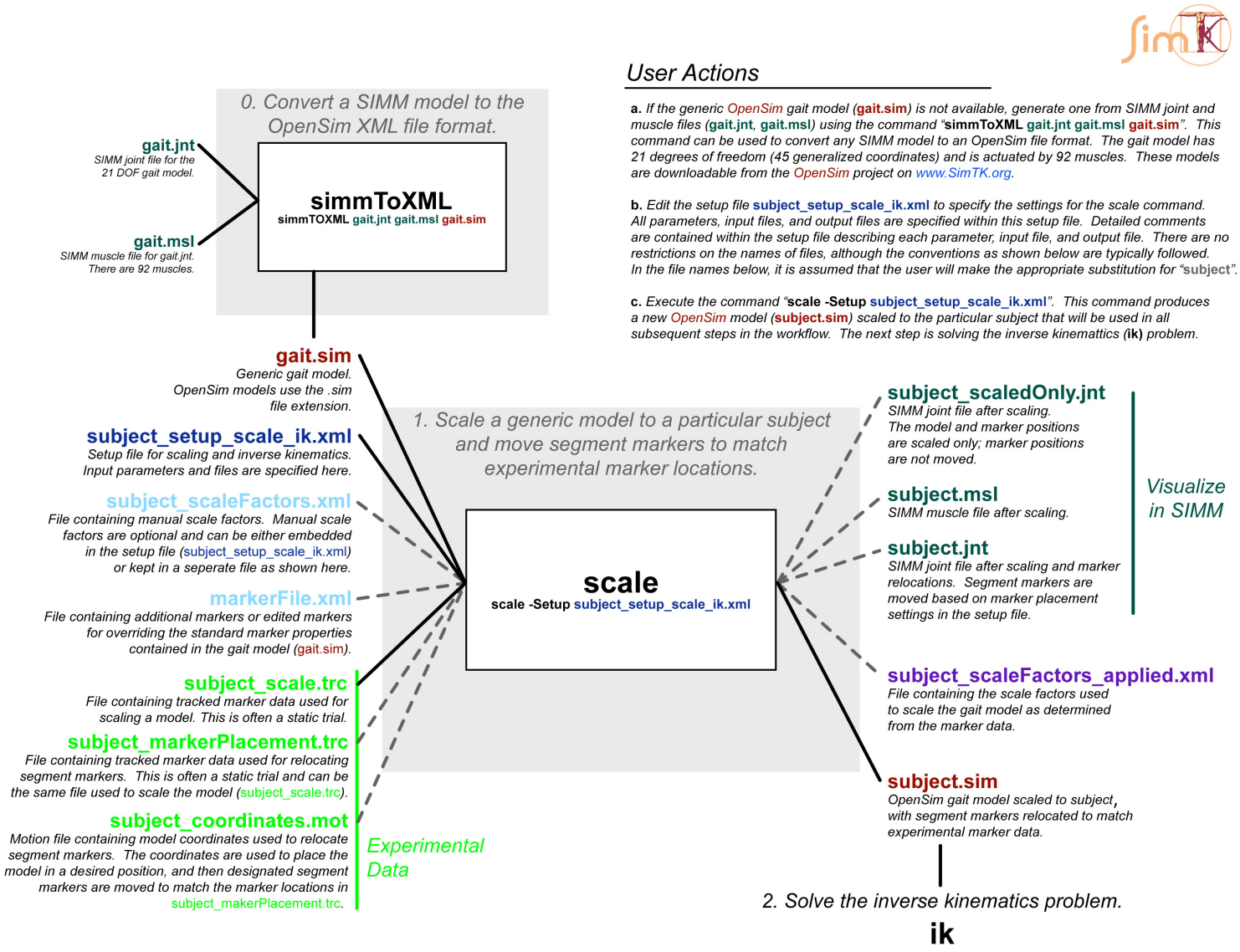

Figure 1.2: Schematic showing the scale step of the Gait Workflow.

User Actions

a. *Edit the setup file* **subject_setup_scale_ik.xml** *to specify the settings for the* **ik** *command. All parameters, input files, and output files are specified within this setup file. Detailed comments are contained within the setup file describing each parameter, input file, and output file. There are no restrictions on the names of files, although the conventions shown below are typically followed. In the file names below, it is assumed that the user will make the appropriate substitution for "***subject***" and "***trial##***".*

b. *Execute the command "***ik -Setup subject_setup_scale_ik.xml***". This command generates a file containing the time histories of joint angles that best reproduce the experimental marker data. This file will be used in the subsequent residual reduction algorithm (***rra***) step.*

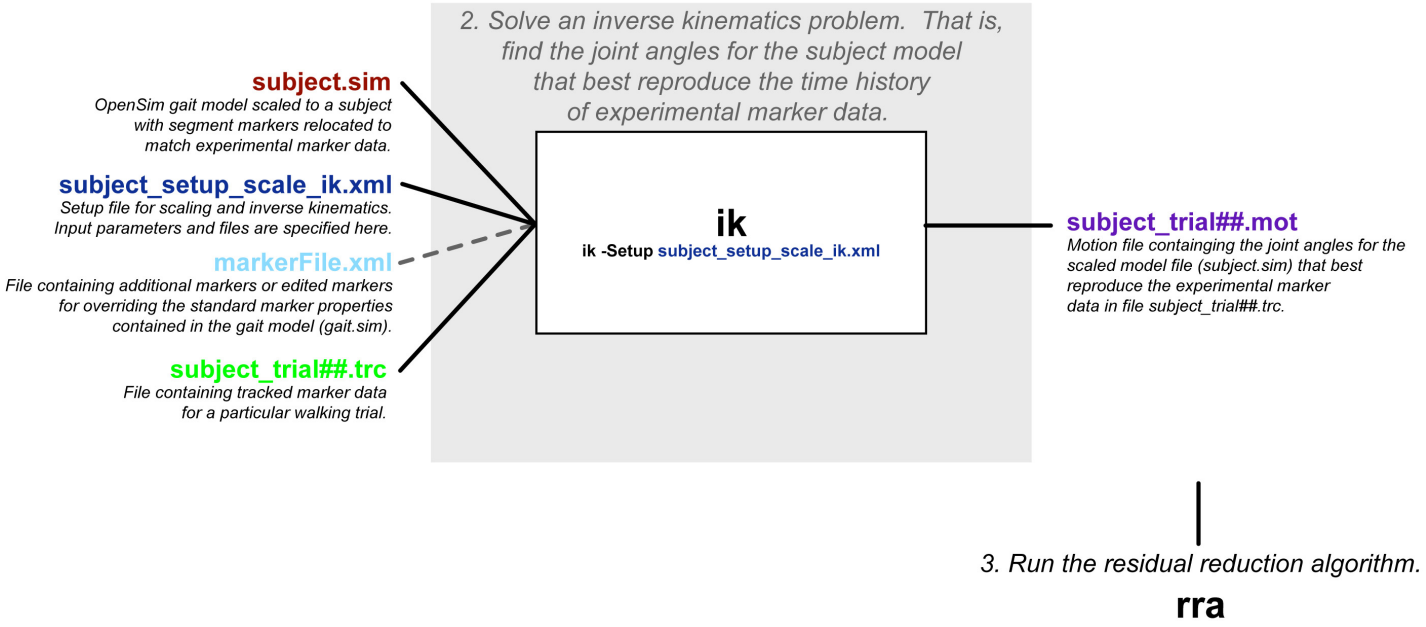

Figure 1.3: Schematic of the ik step of the Gait Workflow.

User Actions

a. *Edit the setup file* **subject_setup_scale_ik.xml** *to specify the settings for the ik command. All parameters, input files, and output files are specified within this setup file. Detailed comments are contained within the setup file describing each parameter, input file, and output file. There are no restrictions on the names of files, although the conventions shown below are typically followed. In the file names below, it is assumed that the user will make the appropriate substitution for "***subject***" and "***trial##***".*

b. *Execute the command "***rra -Setup subject_setup_scale_ik.xml***". This command produces a file containing altered generalized coordinates (q's) so that the motion of the model is more dynamically consistent with the ground reaction data (***subject_trial##_q.sto***). In general, it is not possible for RRA to eliminate residuals entirely. Therefore, the* **rra** *command produces an additional file containing the remaining residuals that need to be applied to the model (***subject_trial##_residuals.sto***). Both these files will be used in the subsequent computed muscle control (***cmc***) step.*

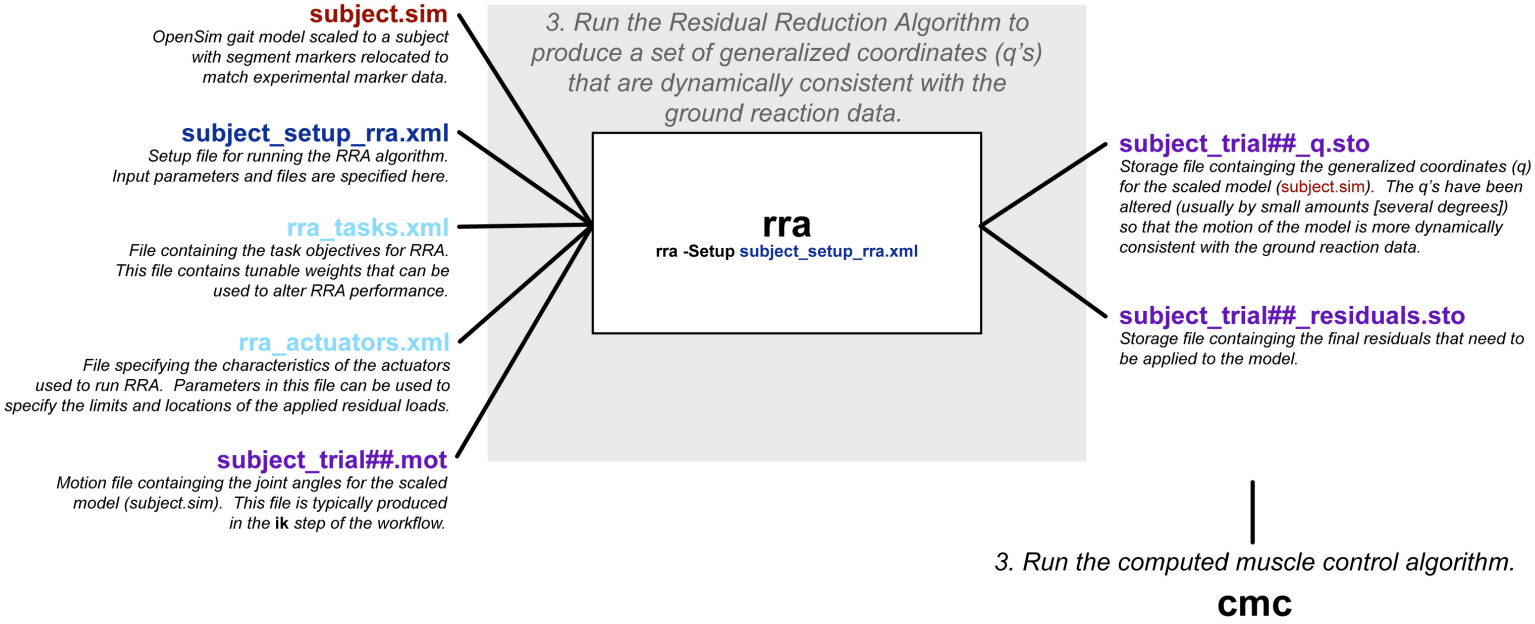

Figure 1.4: Schematic showing the rra step of the Gait Workflow.

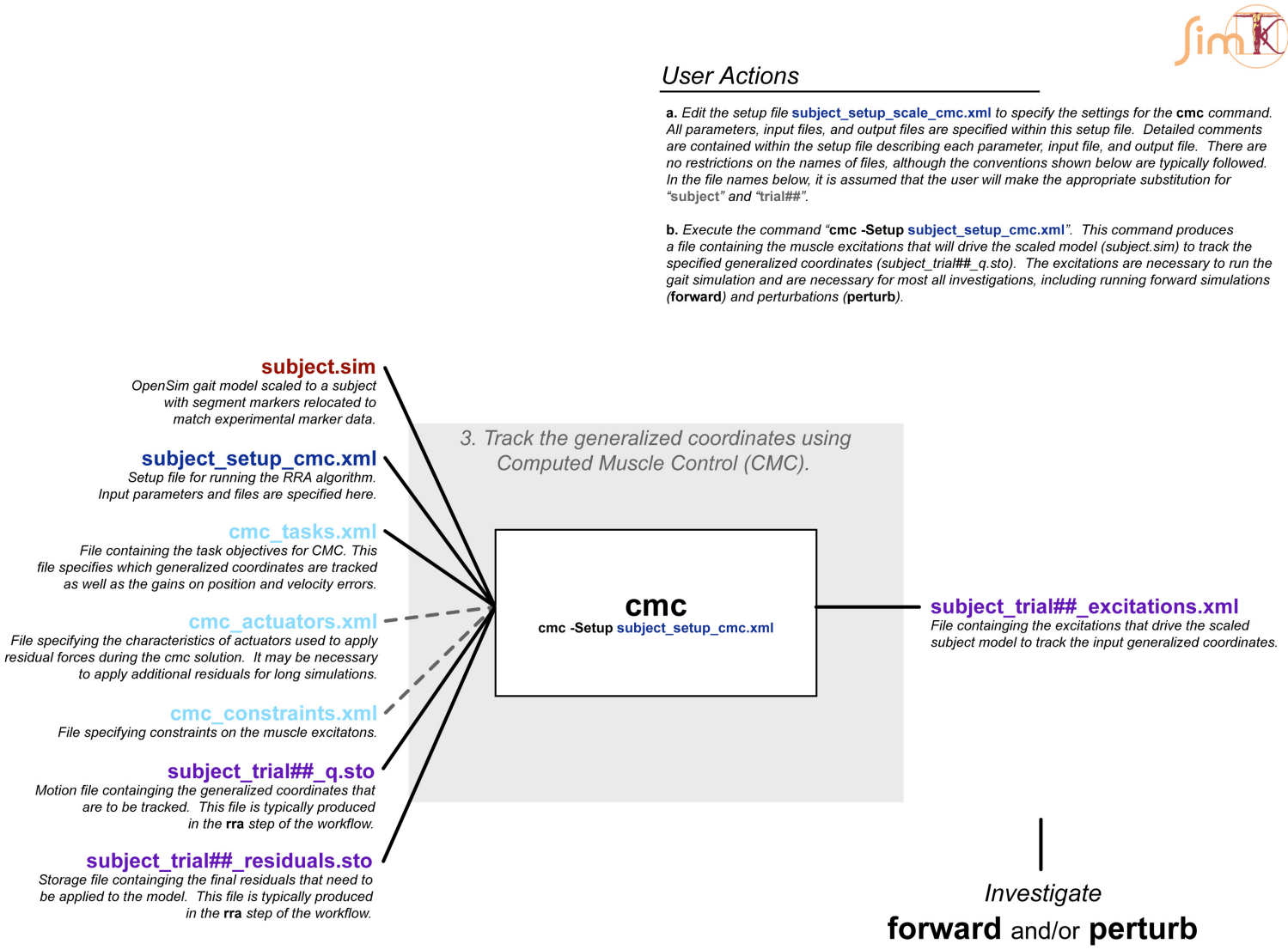

Figure 1.5: Schematic of the cmc step of the Gait Workflow.

Chapter 2

Scaling

The purpose of the `scale` step is to alter the anthropometry of a generic gait model so that it matches a particular subject (Fig. 1.2). The scaling is performed based on experimentally tracked x-y-z marker locations. The marker locations are usually obtained using motion capture equipment. The generic gait model has a set of virtual markers placed in the same locations as the markers commonly used in gait analysis. The dimensions of the segments in the gait model are scaled to achieve the best correspondence between the experimental marker locations and the virtual marker locations.

Tutorial

In this tutorial you'll learn:

1. The structure of input files used by OpenSim
2. How to scale a nominal model based on a static trial.
3. How to scale a model based on precomputed scale factors.

Steps

1. Open the file ***leg.sim*** in the folder OpenSim/Documentation/Tutorials/Scale using your favorite text editor. This file represents a sample model file. All OpenSim's files are written in XML. XML (Extensible Markup Language) is widely-used text-based file format that allows for extensibility. XML files are hierarchical in nature and consist of "elements" that may contain more child "elements". Each element has an openning tag (e.g. <gravity>) and a properly nested, matching closing tag (e.g. </gravity>). The model file ***leg.sim*** has the contents of both the joint and muscle files of SIMM in one file. If you view the file ***leg.sim*** in Internet-Explorer or any other browser, you'll see the file contents represented as a tree that you can collapse/expand as needed to explore the structure of the file.
2. Leave ***leg.sim*** open and open the file ***leg.jnt***, which is the old style SIMM joint file that describes the same model. Search for the block

representing the body/segment ***"pelvis"*** in both leg.jnt and leg.sim. Observe how each attribute of the old format maps to an XML element in the new file format. Display-related attributes are not included in ***leg.sim***.

3. To go through the subject specific simulation pipeline you need a nominal model, that will be scaled to match your subject. To perform this scaling, we need a setup file. The syntax of this file is described in the user manual. However, there's an easy way to generate a default setup file; simply run the command (***scale –PrintSetup***) to generate a file ***default_subject.xml*** in your working directory. Inspect the file in your text editor. For this tutorial a prepared subject file ***subject001_setup_scale_ik_default.xml*** has been provided.
4. When scaling a model, the user has either a recorded trial with a set of markers (static trial, or walking trials) or he/she knows beforehand the scale factors to be applied in X, Y, and Z directions, computed using some other external means. We'll explore both options in this tutorial.
    a. Open the file ***subject001_setup_scale_ik_default.xml*** and edit the contents of the <scalingOrder> element to read **<scalingOrder>measurements</sclaingOrder>** to indicate using measurements from a static trial.
    b. Insert the following lines (or cut and paste from below between the openning <MeasurementSet> and closing </MeasurementSet> tags. This block states that you want to scale the pelvis of the model in the X direction by the average of the measured distance between the two markers (ASIS, Sacral) on left & right sides.

```
<measurement name="pelvis depth">
 <apply>true</apply>
 <MarkerPairSet>
    <markerPair>
                    <markers>R.ASIS V.Sacral</markers>
        </markerPair>
    <markerPair>
                    <markers>L.ASIS V.Sacral</markers>
        </markerPair>
 </MarkerPairSet>
 <BodyScaleSet>
        <bodyScale name="pelvis">
                    <axes>X</axes>
        </bodyScale>
 </BodyScaleSet>
</measurement>
```

    c. You'll need to specify the static pose file (subject001_static.trc) that contains recorded marker data and the time range in this file to use for averaging marker locations as the distance between markers changes with motion:
    <markerFile> subject001_static.trc </markerFile>
    <timeRange> 0.0 0.25 </timeRange>

d. Also change the name of the post scaling SIMM joint and muscle files to read
<outputJointFile> subject001_meas.jnt </outputJointFile>
<outputMuscleFile> subject001_meas.msl </outputMuscleFile>

e. Now save the file ***subject001_setup_scale_ik_default.xml*** as ***mysubject001_setup_scale_ik_meas.xml.***

5. Run the scaling application as "scale –S mysubject001_setup_scale_ik_meas.xml". This will scale the model (nominalModel) using the specified "measurements" and generates the files **subject001_meas.jnt, subject001_meas.msl.**
6. Now launch SIMM and open both pre scaling and post scaling models to verify how the scaling was done.
7. You can use manual scaling instead of using measurements by changing the <scalingOrder> element in the file ***subject001_setup_scale_ik_default.xml***. Use provided scaleSet (subject001_scaleSet_bigfoot.xml) for scaling. This can be done using the following steps:
   a. Change <scalingOrder> element in ***subject001_setup_scale_ik_default.xml*** to **<scalingOrder>manual</sclaingOrder>**
   b. Add (copy/paste) the following lines into the file ***subject001_setup_scale_ik_default.xml*** to refer to the provided scales file <ScaleSet file=" subject001_scaleSet_bigfoot.xml "/>
   c. Rename output jnt and msl files to ***scaleModelManual.jnt***, ***scaleModelManual.msl***.
   d. Save the file mySubject.xml.
8. Run scale again on "mySubject.xml", view output in SIMM.
9. You can actually perform mix-and-match scaling on body segments by doing manual followed by measurements based scaling or vice versa. The scaling methods are applied in the order they appear. For example <scalingOrder> manual measurements </scalingOrder> uses manual scaling first to do scaling then applies measurement based scaling. This can be used when you have high confidence in some specific markers that you'd want to use for scaling but not all markers. Explore with these options for scalingOrder.
10.
11.

SimTK, the Simulation Toolkit, is part of the Simbios National Center Center for Biomedical Computation funded by the National Institutes of Health. The purpose of SimTK is to enable groundbreaking biomedical research by providing open access to high-quality tools for modeling and simulating biological structures.

Chapter 3

Inverse Kinematics (IK)

The purpose of the `ik` step is to find the joint angles in the model that best reproduce the kinematics recorded for a particular subject.

r chapters in the User's Guide where more detailed information can be found.

Tutorial

In this tutorial you'll learn:

1. How to setup inverse kinematics problem in OpenSim
2. The various parameters available to tune the IK algorithm.

Steps

1. Open the file ***IKSubject.xml*** in ikTutorial folder of the OpenSim installation using a text editor. This file represents a snapshot of a subject with ScalingParameters, MarkerPlacementParameters and IKParameters all set. Inspect the different elements under the xml tag `<IKParameters>`. Notice that the file is setup to contain potentially multiple trials for the same subject (these would be included in elements labeled `<IKTrial>` under `<IKTrialSet>`). There are two kinds of parameters for IK, one kind is common to all trials (placed under <IKParameters> directly) and those specific to individual trials (placed under <IKTrial>). We'll experiment with both in this tutorial. First we begin with common IK parameters.

2. Run inverse kinematics on the file ***IKSubject.xml*** as is to solve for the best Joint-angles of the (scaled) model that matches the experimental data file ***walk1.trc***. This is done by invoking solveIK –S IKSubject.xml. The input motion file to IK is walk1.trc. This will take a few seconds, and will result in a motion file with the IK solution.

3. Open SIMM, load the model's joint and muscle file post scaling & marker placement (***scaledModel.jnt***, ***scaledModel.msl***) as well as the resulting motion file from step 2 (***walk1.mot***). Note the difference in the location of the pink & blue marker for (L.Asis).

4. Modify the weight of the marker L.Asis to 100.0 (under <IKParameters><markerSet><marker name="L.Asis">), modify the name of the output motion file to ***preferLAsis.mot***, save to a different file ***IKSubjectLAsis.xml*** and rerun solveIK as in step2.

5. Load motion file output from the previous step (***preferLAsis.mot***) into SIMM and notice how the L.Asis marker position changes (the blue and pink markers almost coincide throughout the trial).

6. Open original ***IKSubject.xml*** and modify the contents of <IKParameters>/<CoordinateSet>/<coordinate name="knee_flex_r">/<locked> to false, change output motion file name to ***walkLockKneeR.mot*** and save to a new file ***IKLockKneeR.xml***. This tells the IK solver to use the values for the generalized coordinate "knee_flex_r" from the specified input file rather than solve for it at each frame. This can be useful when you have enough confidence in some generalized Coordinate value that we don't want the IKSolver to change.

7. Run inverse kinematics on the ***IKSubjectLockKneeR.xml*** and load the results into SIMM and observe the difference.

8. Now for individual trials, you can change the range of time to solve for, kinematicsSmoothing, groundReactionSmoothing.

Chapter 4

Residual Reduction Algorithm (RRA)

The purpose of the `ik` step is to find the joint angles in the model that best reproduce the kinematics recorded for a particular subject.

r chapters in the User's Guide where more detailed information can be found.

Tutorial

In this tutorial you'll learn:

3. How to setup inverse kinematics problem in OpenSim
4. The various parameters available to tune the IK algorithm.

Steps

9. Open the file ***IKSubject.xml*** in ikTutorial folder of the OpenSim installation using a text editor. This file represents a snapshot of a subject with ScalingParameters, MarkerPlacementParameters and IKParameters all set. Inspect the different elements under the xml tag `<IKParameters>`. Notice that the file is setup to contain potentially multiple trials for the same subject (these would be included in elements labeled `<IKTrial>` under `<IKTrialSet>`). There are two kinds of parameters for IK, one kind is common to all trials (placed under <IKParameters> directly) and those specific to individual trials (placed under <IKTrial>). We'll experiment with both in this tutorial. First we begin with common IK parameters.

Chapter 5

Computed Muscle Control (CMC)

The purpose of the `ik` step is to find the joint angles in the model that best reproduce the kinematics recorded for a particular subject.

r chapters in the User's Guide where more detailed information can be found.

Tutorial

In this tutorial you'll learn:

5. How to setup inverse kinematics problem in OpenSim
6. The various parameters available to tune the IK algorithm.

Steps

10. Open the file ***IKSubject.xml*** in ikTutorial folder of the OpenSim installation using a text editor. This file represents a snapshot of a subject with ScalingParameters, MarkerPlacementParameters and IKParameters all set. Inspect the different elements under the xml tag `<IKParameters>`. Notice that the file is setup to contain potentially multiple trials for the same subject (these would be included in elements labeled `<IKTrial>` under `<IKTrialSet>`). There are two kinds of parameters for IK, one kind is common to all trials (placed under <IKParameters> directly) and those specific to individual trials (placed under <IKTrial>). We'll experiment with both in this tutorial. First we begin with common IK parameters.

References

Index