Overview

This set of instructions will guide you through producing CMC simulations for a subject walking at a self-selected speed and subject running at 4 m/s. The running experimental data was collected in a previous study by Hamner *et al* [1]. The collection of the walking data is described in the associated manuscript. All simulations were performed in OpenSim 3.3 [2]. If you have further questions, feel free to contact Apoorva Rajagopal (apoorvar@stanford.edu).

Description of downloaded items

File/Folder	Size	Description/Contents
Rajagopal2015.osim	<1 MB	- OpenSim model file
Geometry	3 MB	- Geometry (*.vtp) files needed to visualize model in the GUI; *not* necessary to run simulations
ExpData	5 MB	- Marker data for static trials (used to scale the model and place model markers) - Marker and ground reaction force (GRF) data for a walking and running trial
Scale	2 MB	- *scale_setup_walk_scaleOnly.xml, scale_setup_walk.xml*: Set up file to run the Scale Tool for the walking subject and place model markers, respectively. - *scale_setup_run.xml*: Set up file to run the Scale Tool for the running subject - *subject_scaled_walk.osim, subject_scaled_run.osim*: output models from the Scale Tool
IK	<1 MB	- *ik_setup_*.xml*: Set up files to run the Inverse Kinematics Tool for walking and running - *results_**: folders containing IK outputs
RRA	62 MB	- Each subfolder (walk, run) contains the necessary setup files to run the RRA tool, folders containing reference results, and RRA-adjusted models (more detail below)
CMC	130 MB	- Each subfolder (walk, run) contains the necessary setup files to run the CMC tool, an adjusted, subject-specific OpenSim model (more detail below), and a folder containing reference results
ID	2 MB	- *id_setup_*.xml*: Set up files to run the Inverse Dynamics Tool for walking and running - *results_**: folders containing ID outputs
scripts	<1 MB	- Matlab scripts to generate the walking and running simulations, and associated validation plots; scripts are described in further detail below
scripts/modifyGenericModel	<<1 MB	- API scripts that contain tools to automatically adjust model body segment masses after RRA, scale muscle forces based on subject mass and height, and set other model parameters.
Videos	11 MB	- Video of walking and running simulations in OpenSim

Platform/Environment

The OpenSim GUI can only be run on a Windows machine (virtual machine is ok). The OpenSim libraries and executables can be built from source in multiple environments, including Mac and Linux machines. Detailed instructions for building OpenSim and using the API are available in the OpenSim User's Guide via opensim.stanford.edu.

Instructions

Download: Save the folder with all simulation data and setup files to the computer you plan to use. This directory will be referred to as DOWNLOAD_DIR in these instructions. These instructions detail how to generate our simulations using either the OpenSim-Matlab scripting interface or OpenSim's command line tools. These simulations can be generated in the GUI as well, although in this case, you will need to manually save and/or edit files in later stages (e.g. RRA). Please read the appropriate section below for instructions on generating these simulations either with Matlab (page 2, below) or command line tools and the OpenSim API (page 3).

Generating simulations using the OpenSim-Matlab scripting interface

Set-up:
Instructions to set up the Matlab scripting environment is available at:
http://simtk-confluence.stanford.edu:8080/display/OpenSim/Scripting+with+Matlab.

Generate simulations:

1. Open Matlab and navigate to DOWNLOAD_DIR/scripts.
2. Run the runAll.m script.
   - This is a "do-all" script calls on the file that generates the muscle-driven simulations for walking and running, plots the inverse-dynamics vs. muscle-generated joint moments, and plots the simulated muscle-activity vs. EMG activity.
   - This script assumes that the OpenSim-Matlab scripting interface has successfully been set up.
   - More detailed information about runAll.m is given below in the **Available files** section.

Available files:
runAll.m: This is a "do-all" script calls on the file that generates the muscle-driven simulations for walking and running, plots the inverse-dynamics vs. muscle-generated joint moments, and plots the simulated muscle-activity vs. EMG activity. This script assumes that the OpenSim-Matlab scripting interface has successfully been set up.

runAll.m calls on the following functions/files to generate the walking and running simulations and generate plots using simulation results

1. makeSimulation('[*walk* or *run*]')
2. validationPlots_emg_* [*walk* or *run*]
3. validationPlots_id_* [*walk* or *run*]
4. validation_RRA_CMC_trackingErrors_* [*walk* or *run*]

makeSimulation.m is a function that does the following:

[NOTE: allowed inputs are 'walk' or 'run' depending on which simulation you would like to generate]

1. Uses the OpenSim **Scale** Tool to generate a model that is scaled to the subject anthropometry;
2. Uses the OpenSim **Inverse Kinematics** Tool to compute joint kinematics from the experimental motion capture data;
3. Uses the OpenSim **Residual Reduction Algorithm** to generate a model and a set of smoothed joint kinematics that improve dynamic consistency between the kinematic and the kinetic experimental data (i.e. between the joint kinematics and the experimentally measured ground reaction forces);
   a. This step is repeated several times. At each step, an "out.log" file is generated which contains the suggested model mass changes to improve dynamic consistency. The out.log file is automatically read and the mass changes are distributed amongst the body segments proportionally.

   b. Note: if there is an issue generating the out.log files (e.g. due to restricted read/write permissions), the function will throw an error.
4. Scales the muscle forces based on the mass-height-volume relationship reported by Handsfield *et al*. (2013) and described in the associated paper;
5. Uses the OpenSim **Computed Muscle Control** Tool to compute the muscle-excitations necessary to generate a muscle-driven simulation of gait. This is the computationally longest step in the pipeline;
6. Uses the OpenSim **Inverse Dynamics** Tool to compute the total joint moment necessary to produce the tracked motion.

validationPlots_emg_*.m is a script that does the following:

1. Extracts the muscle activations computed by the CMC Tool;
2. Loads the experimental EMG data
3. Normalizes the experimental EMG data for each muscle to the maximum value for that muscle's simulated muscle activity.
4. Plots the normalized experimental EMG data (light blue) and the simulated muscle activity (dark blue) vs. percent gait cycle. [For the walking simulation, the script also re-aligns the data so that instead of plotting from 25% to 125% of a gait cycle, the data for the second gait cycle (100-125%) is plotted first, followed by the data from the first gait cycle (25% - 100%).]

validationPlots_id_*.m is a script that does the following:

1. Computes the total muscle-generated hip flexion, knee extension, and ankle plantarflexion joint moment from the CMC results.
2. Loads the joint moments computed by the Inverse Dynamics tool.
3. Plots the Inverse Dynamics joint moments (blue) and the muscle-generated joint moment (red) vs. percent gait cycle. [For the walking simulation, the script also re-aligns the data so that instead of plotting from 25% to 125% of a gait cycle, the data for the second gait cycle (100-125%) is plotted first, followed by the data from the first gait cycle (25% - 100%).]
4. Computes the absolute and normalized RMS and peak differences between the ID and muscle-generated joint moments and prints to the console.

validation_RRA_CMC_trackingerrors_*.m is a script that does the following:

1. Computes the tracking errors from IK to RRA, and from RRA to CMC and prints to the console.

Generate simulations using command line tools and C++ API

Set-up: **Build C++ Script to modify model properties**:
Instructions for how to build the files in *scripts/modifyGenericModel* and link them to the OpenSim libraries are available in the Developer's Guide in the OpenSim documentation:
http://simtk-confluence.stanford.edu:8080/display/OpenSim/Developer%27s+Guide

Simulation Pipeline
Note: the downloaded folder contains all the outputs from each of the steps in the simulation pipeline. If you opt to skip any of the steps, then feel free to use the outputs that are currently provided.

Scale: uses the static trial to scale the generic model's body segment dimensions and muscle-tendon unit geometry, fiber lengths, and tendon lengths

1. Use a terminal window to navigate to the Scale folder in the downloaded simulations directory:
   cd DOWNLOAD_DIR/Scale.

2. Run the OpenSim Scale Tool by entering:
   scale –S scale_setup_walk_scaleOnly.xml
   scale –S scale_setup_walk.xml
   scale –S scale_setup_run.xml

3. This will generate the scaled models subject_scaled_walk.osim and subject_scaled_run.osim.

Inverse Kinematics: generate joint kinematics for both motions using the scaled model and motion capture data

1. Navigate to the IK folder in the downloaded simulations directory:
   cd DOWNLOAD_DIR/IK

2. Run the OpenSim Inverse Kinematics Tool by entering:
   ik –S ik_setup_walk.xml
   ik –S ik_setup_run.xml

3. This will generate motion (.mot) files containing joint kinematics

Residual Reduction Algorithm (RRA): the algorithm is run several times to iteratively generate a model and joint kinematics that together reduce the dynamic inconsistency between measured kinematic and kinetic data [2].

1. Navigate to the RRA folder in the downloaded simulations directory:
   cd DOWNLOAD_DIR/RRA

2. The following steps need to be run for the walking and running simulations.
   a. Walking Simulation
      i. Navigate to the 'walk' folder in the downloaded simulations directory:
         cd DOWNLOAD_DIR/RRA/walk
      ii. Run the OpenSim RRA Tool by entering:
         rra –S rra_setup_walk_1.xml
      iii. The RRA Tool will suggest a total mass change to the model, and will distribute that mass change over the bodies in the model. Either manually change the masses in RRA-generated model file, subject_walk_rra1.osim, or edit and run the following section of the C++ code in SetModelParameters.cpp:

```
// WALK RRA ROUND 1
Model* osimModel = new Model("DOWNLOAD_DIR/RRA/walk/subject_walk_rra1.osim");

double rraSuggMassChange = <rra suggested mass change here>;
setMassOfBodiesUsingRRAMassChange(osimModel, rraSuggMassChange);
osimModel->print DOWNLOAD_DIR/RRA/walk/subject_walk_rra1.osim ");
```

      iv. Repeat steps (ii) and (iii) for RRA round 2 using the appropriate set up files and adjusting the masses using SetModelParameters.cpp.

      v. After completing RRA, adjust the optimal force of all muscle-tendon units using the scaled model's mass and height. This can be done by running the following section of code in SetModelParameters.cpp:

```
/* ------------------------------------------------------------
Adjust model forces after RRA completes, use mass-height-volume relationship
------------------------------------------------------------ */
Model* osimModel = new Model DOWNLOAD_DIR/RRA/walk/subject_walk_rra2.osim");
Model* osimModelUpd = new Model("DOWNLOAD_DIR/RRA/walk/subject_walk_rra2.osim");

double heightOriginal = 1.70; // m
double heightNew = 1.83; // m
scaleOptimalForceSubjectSpecific( osimModel, osimModelUpd, heightOriginal,  heightNew );
setMaxContractionVelocityAllMuscles(osimModelUpd,15.0);

osimModelUpd->print("DOWNLOAD_DIR/CMC/walk/subject_walk_adjusted.osim");
```

Alternatively, for this subject, the force scale factor for all muscles is 1.0643. This can be manually done for all muscles by editing the model file. If you want to skip this, the CMC folder contains a model, subject_walk_adjusted.osim, that the muscle-force adjustment already done.

b. Running Simulation
   i. Navigate to the 'run' folder in the downloaded simulations directory:
      cd DOWNLOAD_DIR/RRA/run
   ii. Run the OpenSim RRA Tool by entering:
      rra –S rra_setup_run_1.xml
   iii. The RRA Tool will suggest a total mass change to the model, and will distribute that mass change over the bodies in the model. Either manually change the masses in RRA-generated model file, subject_run_rra1.osim, or edit and run the following section of the C++ code in SetModelParameters.cpp:

```
// RUN RRA ROUND 1
Model* osimModel = new Model("DOWNLOAD_DIR/RRA/run/subject_run_rra1.osim");

double rraSuggMassChange = <rra suggested mass change here>;
setMassOfBodiesUsingRRAMassChange(osimModel, rraSuggMassChange);
osimModel->print DOWNLOAD_DIR/RRA/run/subject_run_rra1.osim ");
```

   iv. Repeat steps (ii) and (iii) for RRA rounds 2 and 3 using the appropriate set up files and adjusting the masses using SetModelParameters.cpp.

   v. After completing RRA, adjust the optimal force of all muscle-tendon units using the scaled model's mass and height. This can be done by running the following section of code in SetModelParameters.cpp:

```
/* ---------------------------------------------------------------
Adjust model forces after RRA completes, use mass-height-volume relationship
--------------------------------------------------------------- */
Model* osimModel = new Model DOWNLOAD_DIR/RRA/run/subject_run_rra3.osim");
Model* osimModelUpd = new Model("DOWNLOAD_DIR/RRA/run/subject_run_rra3.osim");

double heightOriginal = 1.70; // m
double heightNew = 1.78; // m
scaleOptimalForceSubjectSpecific( osimModel, osimModelUpd, heightOriginal,  heightNew );
setMaxContractionVelocityAllMuscles(osimModelUpd,15.0);

osimModelUpd->print("DOWNLOAD_DIR/CMC/run/subject_run_adjusted.osim");
```

Alternatively, for this subject, the force scale factor for all muscles is 1.0386. This can be manually done for all muscles by editing the model file. If you want to skip this, the CMC folder contains a model, subject_run_adjusted.osim, that the muscle-force adjustment already done.

Computed Muscle Control (CMC): calculate muscle excitations necessary to drive the walking and running motions [3].

1. Navigate to the CMC folder in the downloaded simulations directory:
   cd DOWNLOAD_DIR/CMC

2. The following steps need to be run for the walking and running simulations.
   a. Walking Simulation
      i. Navigate to the 'walk' folder in the downloaded simulations directory:
         cd DOWNLOAD_DIR/CMC/walk
      ii. Run the OpenSim Computed Muscle Control Tool by entering:
         cmc –S cmc_setup_walk.xml
   b. Running Simulation

i. Navigate to the ‘run’ folder in the downloaded simulations directory:
cd DOWNLOAD_DIR/CMC/run
ii. Run the OpenSim Computed Muscle Control Tool by entering:
cmc –S cmc_setup_run.xml

3. This will generate a set of files in the Results directories containing muscle activations and forces, joint kinematics, and joint kinetics.

Inverse Dynamics (ID): calculate total joint moments necessary to produce a specified motion.

1. Navigate to the ID folder in the downloaded simulations directory:
cd DOWNLOAD_DIR/ID

2. Run the OpenSim Inverse Kinematics Tool by entering:
id –S id_setup_walk.xml
id –S id_setup_run.xml

3. This will generate storage (.sto) files containing joint moments.

References

[1] S. R. Hamner and S. L. Delp, “Muscle contributions to fore-aft and vertical body mass center accelerations over a range of running speeds,” *J. Biomech.*, vol. 46, pp. 780–787, 2013.

[2] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, “OpenSim: open-source software to create and analyze dynamic simulations of movement,” *IEEE Trans Biomed Eng*, vol. 54, no. 11, pp. 1940–1950, 2007.

[3] D. G. Thelen, F. C. Anderson, and S. L. Delp, “Generating dynamic simulations of movement using computed muscle control,” *J. Biomech.*, vol. 36, no. 3, pp. 321–28, Mar. 2003.