OpenSim

Getting Started with the Running Challenge

Added by Jennifer Hicks, last edited by Thomas Uchida on May 12, 2014

Download the Running Challenge files here

Provided files

We provide you with two folders (one for each of two subjects) containing the folders and files listed below.

Filename	Description
subjectXX_speedX_cycleX.osim	Model files based on the gait2392_simbody model. Each model has been scaled to match the geometry and mass properties of the subject, and has been adjusted by the Residual Reduction Algorithm (RRA) for each running cycle. We have also added a whole-body metabolic probe that will report the sum of the instantaneous energy expended by all muscles in the model.
gait2392_CMC_Actuators.xml	Specifies the residual and reserve actuators added to the model by the Computed Muscle Control (CMC) Tool.
gait2392_CMC_Tasks.xml	Specifies the coordinates that should be tracked by CMC and the relative importance of each tracking task.
speedX/ExperimentalData /subjectXX_speedX_cycleX_Kinematics_q.sto	OpenSim storage file containing the kinematic data we wish to track. These data were calculated from kinematics measured experimentally and adjusted by RRA.
speedX/ExperimentalData/subjectXX_speedX_GRF.mot	OpenSim motion file containing the ground reaction forces and torques collected experimentally.
speedX/subjectXX_speedX_cycleX_ControlConstraints.xml	Specifies constraints on the model actuators (i.e., muscles and residual/reserve actuators) that must be respected by the CMC optimizer.
speedX/subjectXX_speedX_cycleX_GRF.xml	Specifies the external forces and torques that are applied to the model, and the point at which these forces and torques are applied.
speedX/subjectXX_speedX_cycleX_SetupCMC.xml	Setup file used to configure the CMC Tool.

These files are based on the data collected and analyzed by Sam Hamner (see https://simtk.org/home/runningsim and http://dx.doi.org/10.1016/j.jbiomech.2010.06.025). We provide you with experimental data from two running cycles for each subject (08 and 19) at each speed (4 m/s and 5 m/s). Since your device will be evaluated based on its performance using the cycles you are provided as well as two cycles you are *not* provided, we suggest you use one cycle at each speed for designing ("training data") and reserve one cycle at each speed for confirming that your design will work in general ("testing data"). Please see the Getting Started with CMC Confluence page for more detailed descriptions of the files used by the CMC Tool.

Running CMC

To run CMC in the OpenSim GUI, load the subject-specific model for the gait cycle of interest and launch the CMC Tool. Next, load the corresponding CMC setup file, verify that all the file paths are correct, and click "Run". If you forget which model you have loaded, right-click on the model name in the GUI and select "Info" from the drop-down menu.

You can also run CMC from the command line as follows:

```
cmc -S subjectXX_speedX_cycleX_SetupCMC.xml
```

See the "Best Practices and Troubleshooting Tips" section of the Getting Started with CMC page for helpful tips on using CMC.

Since CMC can take a long time to run, consider doing a test run with a reduced final_time to ensure CMC will be reporting what you need.

Computing metabolic cost

The provided model includes a "whole-body" metabolic probe, which reports the total metabolic cost of all muscles in the model (see the files included with CMC results ending in "_ProbeReporter_probes.sto").The probe reports instantaneous metabolic *power*, so you will need to integrate the resulting time series to determine the total amount of *energy* consumed during the simulation. Please see the Umberger2010MuscleMetabolicsProbe doxygen page for descriptions of the properties associated with the probe. (Note, in particular, the "report_total_metabolics_only" property.)

A Python script for adding whole-body probes to a model is provided in the Simulation-Based Design to Reduce Metabolic Cost example (follow the link at the top of that page to download a .zip file and see "addMetabolicProbes.py"). You can modify this script if you wish to include only a subset of muscles in the probe calculation; you can also use the Matlab scripting interface or modify the .osim file directly.

Since the metabolic probe calculation doesn't affect the results of a simulation (i.e., the probed values are strictly outputs), you can compute metabolics on existing CMC results using the Analyze Tool. You may find this approach useful if you wish to attach additional probes to a model after you've run CMC (the Analyze Tool doesn't run a simulation so it executes substantially faster than CMC).

Adding actuators

The "Scripts" folder provided in the Simulation-Based Design to Reduce Metabolic Cost example contains several scripts for adding actuators. You can use and modify these scripts to get started designing passive assistive devices. If you wish to design an active device, you will need to write a controller; the Creating a Controller example demonstrates adding a controller to the tug-of-war example.

The Simulation-Based Design to Reduce Metabolic Cost example is a great resource in general if you're having trouble getting started.

On your mark, get set, go!