Project: OpenSim Olympics

Added by Thomas Uchida, last edited by Thomas Uchida on May 20, 2014

- Overview
- Grading
- Option A: Standing Long Jump Challenge
  - A.1 - Prerequisites
  - A.2 - Materials Provided
  - A.3 - Requirements and Deliverables
  - A.4 - Main Competition Judging Criteria
  - A.5 - Extensions
- Option B: Four-Minute Mile Challenge
  - B.1 - Prerequisites
  - B.2 - Materials Provided
  - B.3 - Requirements and Deliverables
  - B.4 - Main Competition Judging Criteria
  - B.5 - Extensions

Overview

Recent advances in robotics and musculoskeletal modeling and simulation are enabling us to design and build transformative technology to augment human performance. In the *OpenSim Olympics*, you will design devices using the OpenSim modeling platform and simulate their effects on endurance, strength, and power using musculoskeletal simulations. Upon successfully completing this project, you will be able to:

- Use simulation to study muscle coordination during jumping or running;
- Investigate the effects of augmenting the human body with an idealized assistive device; and
- Refine and optimize assistive device designs using numerical simulation.

Each student will compete in one of the following two challenges (events), described in detail below:

1. **Standing Long Jump:** Augment a musculoskeletal model to maximize long jump distance while achieving a stable landing. You will write C++ code using the OpenSim API to optimize actuator control signals.
2. **Four-Minute Mile:** Augment a musculoskeletal model to minimize metabolic cost during high-speed running. You will edit an OpenSim model file (.osim) or write a Matlab or Python script that adds your assistive device to an OpenSim model and (optionally) write C++ code to control the active components of your device.

You will be given a model and simulation framework to help you get started. The *Qualification Round* will make sure you're on the right track; the *Main Competition* will pit your device against those designed by other competitors. More details about the rules, requirements, and deliverables for each project are described below.

Grading

The project component of this course is worth 50% of your final grade, allocated as follows:

- 10% – *Qualification Round* performance and deliverables (due 5/13).
- 15% – *Main Competition* performance and presentation (due 5/29).
- 25% – Final deliverables: software, report, and video (due 6/5).

Your success in the challenge (maximizing jump distance or minimizing the metabolic cost of running) is only a small portion of your grade. The majority of your grade will be based on how you present your biomechanical design approach and analyze your results. Collaboration with your classmates is encouraged, but each student must submit his or her own deliverables. Please acknowledge and describe the contributions of all collaborators.

Option A: Standing Long Jump Challenge

The **goal of this challenge** is to design a device or suit that will maximize jump distance for a standing long jump. The standing long jump is a historic Olympic event, as well as one of the tests that make up the NFL combine (see this YouTube video, for example). You will be given a planar, torque-driven model with torque limits on the motors that are based on physiological data (i.e., limits on the magnitude and rate of force production). Competitors will additionally receive a framework for performing a dynamic optimization in OpenSim to maximize jump distance while achieving a stable landing (i.e., the position of the model center of mass with respect to the feet must be within a threshold observed experimentally). You must augment the model to maximize jump distance while maintaining this stable landing.

A.1 - Prerequisites

1. Prior C++ programming experience is strongly recommended.
2. Some prior experience with optimization will be helpful.
3. Access to sufficient computational resources for performing optimizations (which can take several hours on a typical desktop computer).
    - You can use the barley FarmShare cluster, which is free for students and with which we have some experience. Please contact Chris Dembia if you're interested in this option.

A.2 - Materials Provided

1. An OpenSim musculoskeletal model for executing a standing long jump. The model is planar with 6 segments (torso, upper arm, lower arm, thigh, shank, and foot), 7 degrees of freedom, and 4 physiological torque motors. Contact is represented with constraints between the foot and ground that can be activated and de-activated.
2. Source code for running an optimization to solve for actuator control signals that maximize jump distance and land successfully (the unassisted case). The source code is written in C++ and uses version 3.2 of the OpenSim API. See the Resources section of the main course Confluence page for helpful links.
3. The objective function in the optimization determines a performance value for each jump (i.e., jump distance penalized by a poor center of mass position at landing and taking advantage of ligament forces at the joints). This objective function will serve as the score for judging the success of your assisted jump.
4. Instructions for setting up your build environment, getting started with simulations/optimizations, and calculating performance. See Getting Started with the Long Jump Challenge.

A.3 - Requirements and Deliverables

Qualification Round

1. Build the project source code in your development environment and run the optimization for an unassisted jump. This optimization will yield a set of torque actuation control signals. Specific suggestions for parameters to test this part are included in the last section of Getting Started with the Long Jump Challenge.
2. Next, add CoordinateActuators at the hip, knee, and ankle, and find values for added torques (via hand-tuning) that increase the model's jump distance. The CoordinateActuators' maximum applied torque must not exceed 100 Nm (the optimal_force property). A stable landing is not required in the Qualification Round.
3. Submit a short report (1–2 pages) containing:
    a. Your jump performance (the total objective function value and the value of each component of the objective);
    b. A plot and brief description of the torques applied by your CoordinateActuators;
    c. A description of your proposed design strategy for the Main Competition;
    d. Your plan for completing the project (including a schedule); and
    e. Acknowledgments.
4. Submit a video of your jump, created by playing back the simulation in the OpenSim GUI.

Main Competition

1. Design and implement a device in OpenSim that works in concert with the human to maximize jump distance while achieving a stable landing. The device may be any combination of OpenSim components. You may also implement new components, as long as you review these with the teaching team. You will use optimization to find the control signals for the human and device actuators, as well as the parameters of your device (if desired).
2. Submit the following files:
    a. Optimization source code (.cpp);
    b. OpenSim model file (.osim) with your device added;
    c. The controls (.xml) for all actuators;
    d. The initial conditions for the model (.sto) that result in your best jump; and

    e. A "readme" file briefly describing each file in your submission.
3. Prepare and submit a report (≤5 pages) containing the following sections:
    a. A description of your device, including diagrams and figures, as needed;
    b. A description of your optimization framework;
    c. A summary and analysis of your results, including a description and plots of the applied forces from your device and how the device affects the human;
    d. Suggestions and feedback for the teaching team; and
    e. Acknowledgments.
4. Record and upload a 2–3 minute video demonstrating your final results. The video should show your best jump and explain how your device works. The video can combine OpenSim animations, figures, and plots with narrative voiceover and/or descriptive text (see the video on this page for an example). You may submit a blooper reel as well, if you wish.

A.4 - Main Competition Judging Criteria

The winning competitor will achieve the optimal objective function value. The objective function is based on the jump distance and penalties for the following:

- The center of mass position with respect to the feet at landing (ensures the landing is stable); and
- Taking advantage of coordinate limit forces at the joints (prevents injury).

You may change the objective function when designing and optimizing your device, but the final score will be computed using the original objective function distributed with the project. In the case of a tied score, the device with the lowest peak power will be the winner. Note that the assistive device is assumed to be massless.

The winning human+device model and simulation must also meet the following **competition rules**:

1. The device may include any combination of model components in OpenSim, either active or passive. You may also create new OpenSim components, if you review the proposed design for any new model components with the teaching team and submit the source code for review.
2. The per joint torque limit for a device is 200 Nm.
3. Competitors may not change any properties of the human model, but they may change the muscle controls, the optimization framework (e.g., objective function or optimizer), or controls parameterization (e.g., number of control points).
4. The model's feet must leave the ground within 10 seconds of the start of the simulation and can leave the ground only once.
5. Passive devices must have no stored energy at the beginning of the simulation.
6. The device must not connect or contact the floor. Only the provided foot-floor constraints may exert forces on the ground.
7. The teaching team may add rules, if necessary, to maintain the integrity of the competition.

A.5 - Extensions

If you finish your device design and jump optimization early, you can extend your work in several ways:

- Implement geometry to visualize your device and/or improve the geometry of the jumper.
- Design a strictly passive device.
- Design a landing controller.
- Test your device with different model anthropometries (e.g., total body mass, mass distribution, or segment lengths).


To learn more and get started, see the Getting Started with the Long Jump Challenge page.

Option B: Four-Minute Mile Challenge

The **goal of this challenge** is to design a device that will minimize the energy consumed by muscles during high-speed running. The ultimate vision is to build a device or suit that will enable an individual who is capable of running a five- or six-minute mile to run a four-minute mile, a threshold for elite performance (as archived in this YouTube video). You will be given subject-specific, three-dimensional musculoskeletal models, OpenSim motion data (i.e., kinematics and ground reaction force profiles) collected from these subjects while running at 4 and 5 m/s, and setup files for generating muscle-driven simulations using the Computed Muscle Control (CMC) Tool. Your device and control strategy will be evaluated using the given data as well as (similar) data collected from the same subjects

running at the same speeds (i.e., additional experimental trials).

B.1 - Prerequisites

1. Prior Matlab or Python scripting experience is strongly recommended, along with C++ programming experience to create a controller for any active devices.
2. Some prior experience with optimization and/or the biomechanics of human performance will be helpful.
3. Access to sufficient computational resources for performing optimizations or iterating on device designs.

B.2 - Materials Provided

1. A generic three-dimensional OpenSim model for simulating human running that has been scaled to two specific subjects. The models are based on the gait2392_simbody model (described on the Gait 2392 and 2354 Models Confluence page and in Hamner et al., 2012). Modifications include the addition of torque-driven arms and metabolic probes for calculating the cost of locomotion.
2. Experimental data for running at 4 and 5 m/s for these two individuals, along with setup files to run CMC and generate muscle-driven simulations that calculate metabolic cost. One of the trials provided for each subject at each speed should be used as "training" data to design your device; the other should be used as "testing" data to evaluate the performance of your design on new motion data.
3. A script to help calculate the performance of your device.
4. Instructions for getting started with running simulations and calculating performance. See Getting Started with the Running Challenge.

B.3 - Requirements and Deliverables

Qualification Round

1. Run CMC in OpenSim for each of the provided subject/trial combinations (2 subjects × 2 speeds × 2 trials). Calculate the total metabolic cost for each trial and identify the 10 muscles with the greatest total energy consumption.
2. Make it "cheaper" for the optimizer in CMC to use one of the model's reserve actuators (hip, knee, or ankle) to track the motion (vs. tracking the motion by applying muscle forces) by changing the corresponding reserve actuator's optimal_force. The reserve actuator then acts like an ideal assistive device. Determine the change in total metabolic cost for all four training simulations (1 trial for each subject at each speed).
3. Submit a short report (1–2 pages) containing:
    a. Figures showing the top consumers of metabolic energy and the metabolic cost before and after adding assistance;
    b. A plot of the torques applied by the cheap reserve actuators;
    c. A description of your proposed design strategy for the Main Competition;
    d. Your plan for completing the project (including a schedule); and
    e. Acknowledgments.
4. Submit a video of one of your running simulations.

Main Competition

1. Design and implement a device (and controller, if your device contains active components) in OpenSim that works in concert with the human to minimize metabolic cost during high-speed running. The device may consist of any combination of OpenSim components and should be controlled by functions of system states (e.g., knee joint angle) rather than being a function of time. Such a control strategy will enable your device to work effectively with the new trials that will be used to evaluate your design. You may also design a purely passive device. To evaluate the effectiveness of your device with new motions, we suggest using "training" motion trials to build your device and "testing" motion trials to evaluate performance. You will be penalized for the peak power that the device injects into the system at any point in the simulation. If desired, you may use optimization to tune the design of your device. Competitors will use the CMC Tool to solve for muscle and device actuation.
2. Submit the following files:
    a. OpenSim model file (.osim) for each subject with your device added;
    b. Any source code (.cpp) for new components or optimization; and
    c. A "readme" file briefly describing each file in your submission.
3. Prepare and submit a report (≤5 pages) containing the following sections:
    a. A description of your device and control strategy, including figures, as needed;
    b. A description of the process you used to design your device and controller;
    c. A summary and analysis of your results, including a description/plots of the applied forces from your device and how the device affects the human (e.g., muscle activations and forces);
    d. Suggestions and feedback for the teaching team; and
    e. Acknowledgments.
4. Record and upload a 2–3 minute video demonstrating your final results. The video should show your best design and explain how

your device works. The video can combine OpenSim animations, figures, and plots with narrative voiceover and/or descriptive text (see the video on this page for an example). You may submit a blooper reel as well, if you wish.

B.4 - Main Competition Judging Criteria

The winning competitor will achieve the greatest percent reduction in metabolic cost, with an energetic penalty for the added mass of your device. This performance metric will be averaged across all subjects and trials, including the two given (training and testing) and two completely new/unknown trials selected by the teaching team. When calculating the score for each subject/trial combination, note the following:

- Metabolic cost for each subject/trial is defined as the sum of the metabolic cost for all muscles in the model, averaged over the running gait cycle, calculated using the Umberger metabolic probes attached to the provided models. Percent reduction in metabolic cost from the unassisted case will be averaged across both the given (training and testing) and new/unknown subjects/trials.
- A penalty will be included to account for the mass of the device. We will provide a formula to estimate the mass of your device (assumed to be dominated by the masses of batteries and motors).

Specifically, the Judging Score, ***S***, will be computed as:

$$S = \sum_{trials}\{E_{nominal} - E_{assisted} - C_{running}(\frac{E_{device}}{e_{lithium}} + \sum_{actuators}\frac{M_{peak}}{\tau_{motor}})\}$$

which states that the total score is the sum of scores for all running trials, where:

$E_{nominal}$ = is the metabolic consumption during the trial when assistance is not provided

...

$E_{assisted}$ = is the metabolic consumption during the trial when assistance is provided

...

$C_{running}(\frac{E_{device}}{e_{lithium}} + \sum_{actuators}\frac{M_{peak}}{\tau_{motor}})$ = estimated cost of transport due to the added mass of the device.

...

$C_{running} = 3.0\frac{J}{kg-m}$, the experimentally measured cost of running 1 meter per kg of body mass

...

E_{device} = total work done by the device actuators

...

$e_{lithium} = 9.5 * 10^5\frac{J}{kg}$, specific energy capacity of lithium ion.

...

M_{peak} = the peak torque applied by an assistive actuator during the trial.

...

$\tau_{motor} = 10Nm/kg$, the specific strength of a typical, off the shelf electric motor.

The winning human+device model and simulation must also meet the following **competition rules**:

1. The device may include any combination of model components in OpenSim, either active or passive. You may also create new OpenSim components, if you review the proposed design for any new model components with the teaching team and submit the source code for review.
2. The mass of the system must be constant throughout the simulation.
3. For the Main Competition, competitors may not change any properties of the human model or any of the existing CMC settings corresponding to the muscles and residual/reserve actuators. (After the Qualification Round, reserve actuator settings must be

returned to the original values.)
4. The device cannot interact with the ground.
5. The teaching team may add rules, as needed, to maintain the integrity of the competition.

B.5 - Extensions

If you finish your assistive device design and optimization early, you can extend your work in several ways:

- Implement geometry to visualize your device.
- Design a strictly passive device.
- Submit a Matlab or Python script that adds your device and controller to a new OpenSim model. The script can take any parameters of the model as input, but cannot access any information about a simulated motion—that is, you will implement a method with a signature "augmentedModel = myDesign(originalModel)".
- Design a device that can assist both walking and running.
- Describe how you would design a human controller (to replace CMC) that would adjust kinematics and ground reaction forces to adapt to the device/suit.

To learn more and get started, see the Getting Started with the Running Challenge page.