OpenSim

Simulation-Based Design to Prevent Ankle Injuries using the OpenSim-Matlab Interface

Overview

The purpose of this exercise is to use the OpenSim-Matlab interface to evaluate the risk of injury during landing and to design assistive devices to prevent injuries. This exercise is an extension to the 'Simulation-Based Design to Prevent Ankle Injuries' tutorial and follows the same structure. However, here we demonstrate how simulations with varying conditions (such as orthosis stiffness) can be performed using the OpenSim-Matlab interface. As a result, simulations can be embedded in user-defined approaches for optimal design of assistive devices in Matlab.

I. Explore the OpenSim libraries and musculoskeletal model

Launch matlab.

To import the OpenSim modeling classes, use the command:

```
>> import org.opensim.modeling.*
```
Read in the osim model using the following command:

```
>> osimModel = Model([Example_Dir\ToyDropLanding
'\ToyLandingModel.osim']);
```
with Example_Dir the path of the location where you copied the ToyDropLanding folder.

You can for example obtain the number of coordinates of the model using getNumCoordinates():

```
>> nDOF = osimModel.getNumCoordinates()
```

Use the methods command to view the methods in the OpenSim Model class:

```
>> methods Model
```

```
>> methods org.opensim.modeling.Model
```

You can obtain a window listing the signature (arguments, return type) of all methods in an OpenSim Class:

```
>> methodsview Model
```

Use this list to find a method that returns a reference to the publications this model is based on.

II. Evaluate ankle inversion injury during a drop landing on a sloped surface

A. Forward simulation to simulate drop landing

Open evaluateAnkleInversionInjury.m and complete the script.

1. Add the path to the example directory [Example_Dir].
2. Find the line of code where a Forward Tool is defined. The run() method of this tool will simulate the movement. Complete the settings in the lines preceding tool.run(). Set the initial and final time for the simulation to 0 and 0.4.
3. Run the script. Investigate the output files. You can load the [outputName]_states.sto file in the GUI to visualize the motion. (In the OpenSim GUI: make sure ToyLandingModel.osim is the current model, select File>Load Motion… and in the dialog box that appears, navigate to the folder where [outputName]_states.sto is stored.)
4. Alternatively, you can set up the visualizer to show the model and simulation while running the Forward Tool in Matlab by adding the command `osimModel.setUseVisualizer(true)` to your script (insert after the command to read in the osim model).

Question

What is the maximum subtalar angle during the drop landing?

B. Inverse kinematic analysis of experimentally recorded drop landing

We can compare the kinematics of the simulated drop landing to the kinematics of the recorded drop landing. Therefore, we will perform an inverse kinematics analysis based on the scaled model for the test subject and the trc-file containing the marker coordinates. Open setupAndRunIK_Example.m and complete the script.

1. Add the command to import the OpenSim modeling classes at the beginning of the script.
2. Complete the missing input and output paths.
3. Set the initial and final times for the Inverse Kinematics Tool. You can use the `methodsview(ikTool)` command to obtain a window listing the signature of all methods of the InverseKinematicsTool class.
4. Add the command to run the Inverse Kinematics Tool as indicated in the script.
5. Run the script.

Question

What was the maximal subtalar angle during the drop landing experiment? How does this compare to the simulated subtalar angle? Which factors can explain the observed differences?

III. Analyze the effects of an ankle-foot orthosis (AFO) on ankle inversion.

A. Simulate and analyze a drop landing with a “soft” AFO

1. Using the ToyLandingModel_AFO.osim model, simulate the drop landing with the default “soft” AFO (tip: use the evaluateAnkleInversionInjury.m script as an example).
2. Add the trajectory of the subtalar angle to the figure we created in part II.

B. Simulate and analyze a drop landing with a “stiff” AFO

Now let’s make the AFO stiffer by editing the properties of the AFO through the OpenSim-Matlab interface. Open evaluateAFOEffectsOnAnkleInjury.m and complete the script (section ‘Simulate drop landing with a “stiff” AFO’).

1. Choose a new stiffness value.
2. Investigate lines 63-65 of the script. The safeDownCast method downcasts an object from an abstract class to a derived class. To which class does `osimModel.getForceSet().get('AFO_med_bushing')` belong?
3. Use the command `methodsview(AFObushingForce_med)` to obtain a window listing the signature of all methods in the Bushing Force class and find a method to set the translational stiffness. Then, complete the command.
4. Now, add code to change the translational stiffness of the lateral bushing force of the AFO.
5. Run the script to simulate the motions with the soft and adapted AFO and compare the resultant subtalar joint angles.

C. Investigate the trade-off between AFO stiffness and deviation from neutral subtalar angle

Write a script that returns the graph of the maximal subtalar angle as a function of AFO stiffness. Based on this graph, which stiffness value would you pick for the AFO?

IV. Create an active orthotic

A. Add an actuator and controller

We will add a torque motor at the ankle to model an active orthotic and a controller that controls this actuator. We will use a spline function to specify the controls. Open createActiveOrthotic.m and complete the script.

1. Read in `ToyLandingModel_AFO.osim`.
2. Set the AFO actuator properties following the instructions on lines 23-26. Investigate line 24. The Model class has no method to directly get a coordinate. Therefore, we first use the getCoordinateSet method to get a Coordinate Set, and then use the get method of the Coordinate Set class to get the ‘subtalar_angle_r’ coordinate.
3. Define the control nodes on lines 39-40.
4. Run the script and evaluate the subtalar angle.

B. Design optimal AFO control for drop landing

Optimize the timing and activation level of the active orthotic to prevent ankle inversion injury. Tip: make use of the available Matlab functionalities.

IV. Analyze the effects of muscle co-activation

A. Simulation with muscle co-activation

Open analyzeMuscleCoactivation and complete the script to enable the controllers that set the baseline activation levels of inverters and everters and analyze the effect of increased levels of co-activation.

1. On lines 19-26, complete the commands to access, to enable and to modify the controls of the inverter controllers.
2. Write code to enable and modify the controls of the everter controllers.
3. Run the script and explore the effect of increased levels of co-activation.

B. Analyze the effect of the reflex gain

If you are up for a more challenging task, write code to modify the gain of the reflex controllers and analyze the effect of the reflex gain on the subtalar angle.

V. Design an optimal device and training program to prevent injury

Repeat the design challenge using the OpenSim-Matlab interface. The rules stipulated in the 'Simulation-Based Design to Prevent Ankle Injuries' tutorial still apply. Does the increased level of automation allow you to come up with an improved design?

Credits

Friedl De Groote, Mariska Wesseling, Ilse Jonkers, Sam Van Rossom, Hans Kainz with input from James Dunne.