Modeling in OpenSim
23 July 2012
J Rogers

External devices may be modeled in OpenSim via XML. Geometry and mass properties are gotten from SolidWorks or other CAD software. Software versions used for the example here are:

SolidWorks Education Edition 2011-2012 / 2011 SP4.0
OpenSim 3.0 Beta 3

Fig. 1: Part in OpenSim model.

In this example a part is added to an OpenSim model of the pelvis-femur, Fig. 1. The part is free to rotate.

Summary of steps:
Determine the coordinate system used by the OpenSim model.
Design the parts in SolidWorks paying attention to the coordinate system.
Find the mass properties from SolidWorks (Evaluate tab > Mass Properties). Find:

Mass
CG location
Inertia matrix about CG

Add bodies, joints, and coordinates via XML to the OpenSim model.

Detailed Instructions:
Find the coordinate system that is used for the OpenSim model. The triad icon in the upper right hand corner toggles the coordinate system display. Create the geometry in SolidWorks or other CAD software. Plan your coordinate system location and orientation. In SolidWorks the z axis is normal to the front plane.

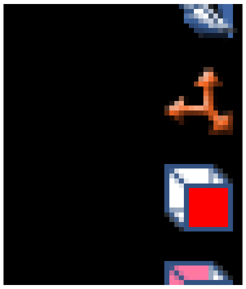

Fig. 2: OpenSimTriad icon

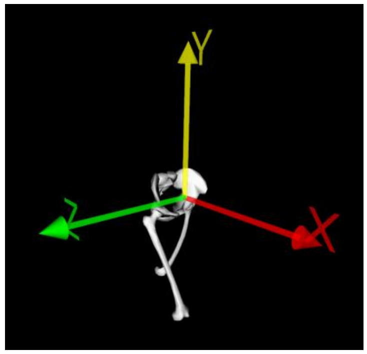

Fig. 3: Pelvis coordinate system

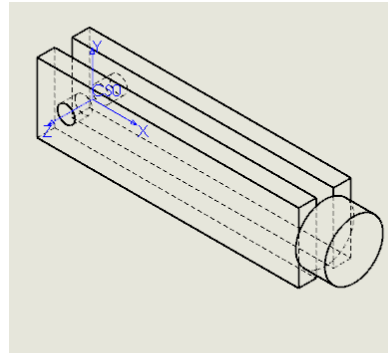

Fig. 4: Part with body coordinate system

Save the SolidWorks part in the usual way and also "Save As,"and select STL (*.stl) for type.
Click the "Options" button and:
Check "Do not translate STL output data to positive space."

The other options should keep their default values (binary output, Meters units, default output coordinate system).

Find mass properties, Fig. 5 (found on the Evaluate tab). Use the options button to set the accuracy and use scientific notation. OpenSim requires the Mass and the inertia matrix elements taken at the center of mass and aligned with the output coordinate system:

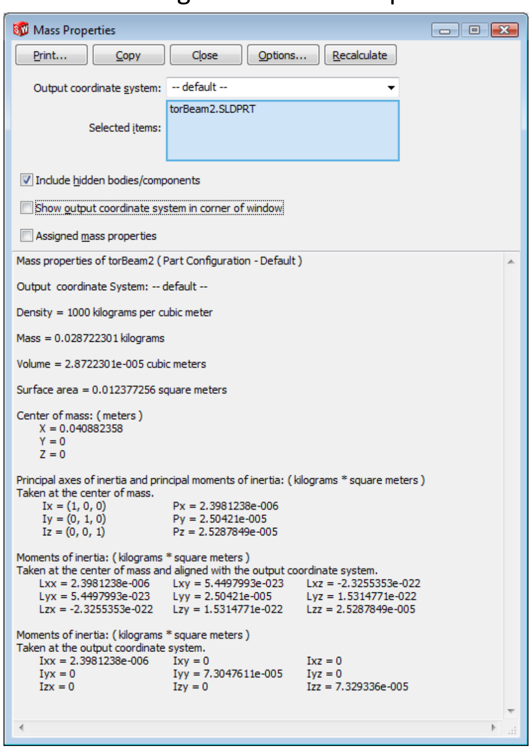

Fig. 5: SolidWorks Mass Properties

Mass = 0.028722301 kilograms
Center of mass: (meters)
X = 0. 0.040882358
Y = 0
Z = 0
Moments of inertia: (kilograms * square meters)
Taken at the center of mass and aligned with the output coordinate system.
Lxx = 2.3981238e-006
Lyy = 2.50421e-005
Lzz = 2.5287849e-005
In this case the cross terms are small enough to ignore, so set these to 0 in OpenSim
Lxy = 5.4497993e-023 (replace with 0)
Lxz = -2.3255353e-022 (replace with 0)
Lyz = 1.5314771e-022 (replace with 0)

Modify or create an OpenSim model via XML.

Add the body with its mass properties, display geometry with the STL file, joints, and coordinates. In this example the part is named “torsionBar_r.

The mass properties are on lines 605-613 in Fig. 6 below. The Pin Joint and Coordinate begin on line 615 in Fig. 6 below. The pelvis is listed as the parent body. The joint “location in parent” (0.020   0   0.15) in this example is given in the pelvis reference frame.

```
14          <!--Bodies in the model.-->
15          <BodySet name="">
16              <objects>
17                  <Body name="ground">
51                  <Body name="pelvis">
282                 <Body name="femur_r">
443                 <Body name="femur_l">
604                 <Body name="torsionBar_r">
605                     <mass> 0.02872230  </mass>
606                     <mass_center> 0.04088236 0 0 </mass_center>
607                     <!--Inertia properties about the center of mass.-->
608                     <inertia_xx> 0.00000240  </inertia_xx>
609                     <inertia_yy> 0.00002504  </inertia_yy>
610                     <inertia_zz> 0.00002529  </inertia_zz>
611                     <inertia_xy> 0           </inertia_xy>
612                     <inertia_xz> 0           </inertia_xz>
613                     <inertia_yz> 0           </inertia_yz>
614                     <!--Joint that connects this body with the parent body.-->
615                     <Joint>
616                         <PinJoint>
617                             <parent_body> pelvis </parent_body>
618                             <location_in_parent> 0.020 0 0.15 </location_in_parent>
619                             <orientation_in_parent> 0 0 0</orientation_in_parent>
620                             <location> 0 0 0</location>
621                             <!-- use 4.712388980384690 for 3/2 * pi -->
622                             <orientation> 0 4.712388980384690 0</orientation>
623                             <!--Generalized coordinates parameterizing this joint.-->
624                             <CoordinateSet>
625                                 <objects>
626                                     <Coordinate name="torsionPinJoint_r">
627                                     <motion_type> rotational</motion_type>
628                                     <!--five degrees: 0.08726646  thirty degrees: 0.52359878
629                                     <default_value> 0.0 </default_value>
630                                     <default_speed_value> 0 </default_speed_value>
631                                     <range> -1 1</range>
632                                     <clamped> false</clamped>
633                                     <locked> false</locked>
634                                     <prescribed_function> No Objects</prescribed_function>
635                                     <prescribed> false</prescribed>
636                                     <is_free_to_satisfy_constraints> false</is_free_to_satisf
637                                     </Coordinate>
638                                 </objects>
639                             </CoordinateSet>
640                             <reverse> false</reverse>
641                         </PinJoint>
642                     </Joint>
643                     <VisibleObject>
```

Fig. 6: XML File showing mass properties, joint, and coordinate

The STL file name is on line 648 in Fig. 7 below:

```
615                 <Joint>
642                 <VisibleObject>
643                     <!--Set of geometry files and associated attributes, allow .vtp
644                     <GeometrySet>
645                         <objects>
646                             <DisplayGeometry>
647                                 <!--Name of geometry file .vtp, .stl, .obj-->
648                                 <geometry_file>torBeam2.stl</geometry_file>
649                                 <!--Color used to display the geometry when visible
650                                 <color> 1 1 1</color>
651                                 <!--Name of texture file .jpg, .bmp-->
652                                 <texture_file />
653                                 <!--in body transform specified as 3 rotations (rad
654                                 <transform> -0 0 -0 0 0 0</transform>
655                                 <!--Three scale factors for display purposes: scale
656                                 <scale_factors> 1 1 1</scale_factors>
657                                 <!--Display Pref. 0:Hide 1:Wire 3:Flat 4:Shaded-->
658                                 <display_preference>4</display_preference>
659                                 <!--Display opacity between 0.0 and 1.0-->
660                                 <opacity>1</opacity>
661                             </DisplayGeometry>
662                         </objects>
663                         <groups />
664                     </GeometrySet>
665                     <!--Three scale factors for display purposes: scaleX scaleY sca
666                     <scale_factors> 1 1 1</scale_factors>
667                     <!--transform relative to owner specified as 3 rotations (rad)
668                     <transform> -0 0 -0 0 0 0</transform>
669                     <!--Whether to show a coordinate frame-->
670                     <show_axes>false</show_axes>
671                     <!--Display Pref. 0:Hide 1:Wire 3:Flat 4:Shaded Can be override
672                     <display_preference>4</display_preference>
673                 </VisibleObject>
674                 <WrapObjectSet>
675                     <objects />
676                     <groups />
677                 </WrapObjectSet>
678             </Body>
679         </objects>
680         <groups/>
681     </BodySet>
682     <!--Constraints in the model -->
```

Fig. 7: XML code showing visible object properties

The default axis of rotation for the pin joint is obtained by aligning the z axis of the torsionBar_r body (the child) with the z axis of the pelvis (the parent). A positive value of the coordinate is a rotation of the body and its coordinate system about the parent z axis according to the right hand rule.

Setting the body and joint orientation other than the default requires that “orientation-in-parent” and “orientation” be set. Figs. 8 and 9 below show the desired rotation of the pin joint. In the example leaving the default “orientation-in-parent” parameter (0 0 0) and changing the “orientation” parameter to (0 4.712388980384690 0) changes the axis such that the joint allows rotation of the torsionBar_r body about a left-to-right axis in the pelvis reference frame with positive coordinate in the same direction as hip flexion for the right side. These two parameters can be used to set any alignment and the positive sense in either direction.

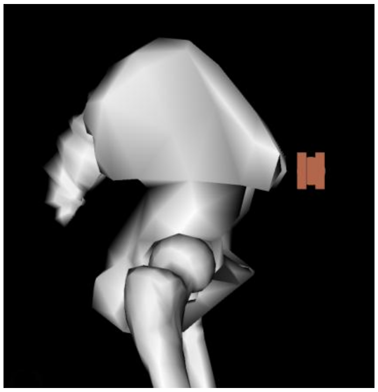

Fig. 8: Zero degrees rotation of the torsionBar_r part

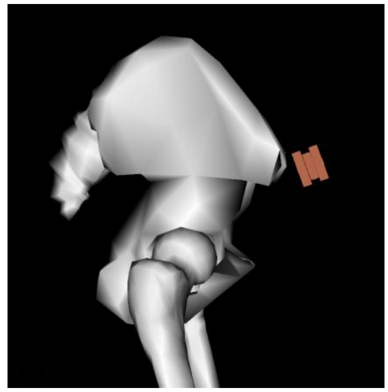

Figure 9 Twenty degrees rotation of the torsionBar_r part