CMC Settings Files and XML Tag Definitions

This section of the chapter covers how to control CMC execution. The properties governing execution are contained in XML files. The topics covered in this section include:

CMC Setup File

Execution of the CMC tool is controlled by properties specified in the CMC setup file. Some properties, such as <initial_time> and <final_time>, are specified directly in the CMC setup file. Other properties, such as <task_set_file>, refer to additional files that contain additional settings that affect CMC execution. These are discussed in the sections that follow.

The properties for CMC are enclosed inside the opening and closing tags <CMCTool> and </CMCTool>. The name attribute name="subject01_walk1" specifies the execution name. The names of results files generated will be prefixed with this name.

Example 1: XML file for the CMC setup file

```
<?xml version="1.0" encoding="UTF-8"?>

<OpenSimDocument Version="20302">

  <CMCTool name="subject01_walk1">
    <!--Name of the .osim file used to construct a model.-->
    <model_file> subject01_simbody_adjusted.osim </model_file>

    <!--Replace the model's force set with sets specified in
        <force_set_files>? If false, the force set is appended to.-->
    <replace_force_set> false </replace_force_set>

    <!--List of xml files used to construct an force set for the model.-->
    <force_set_files> gait2354_CMC_Actuators.xml </force_set_files>

    <!--Directory used for writing results.-->
    <results_directory> ./subject01_ResultsCMC </results_directory>

    <!--Output precision. It is 8 by default.-->
    <output_precision> 20 </output_precision>

    <!--Initial time for the simulation.-->
    <initial_time> 0.76000000 </initial_time>

    <!--Final time for the simulation.-->
    <final_time> 1.24000000 </final_time>

    <!--Flag indicating whether or not to compute equilibrium values for
    states other than the coordinates or speeds. For example, equilibrium
    muscle fiber lengths or muscle forces.-->
<solve_for_equilibrium_for_auxiliary_states> true </solve_for_equilibrium_for_auxiliary_states>

<!--Maximum number of integrator steps.-->
<maximum_number_of_integrator_steps> 30000 </maximum_number_of_integrator_steps>

<!--Maximum integration step size.-->
<maximum_integrator_step_size> 1.00000000 </maximum_integrator_step_size>

<!--Minimum integration step size.-->
<minimum_integrator_step_size> 0.00000000 </minimum_integrator_step_size>

<!--Integrator error tolerance. When the error is greater, the integrator
    step size is decreased.-->
<integrator_error_tolerance> 0.00000100 </integrator_error_tolerance>

<!--Set of analyses to be run during the investigation.-->
<AnalysisSet name="Analyses">
  <objects/>
  <groups/>
</AnalysisSet>

<!--Controller objects in the model.-->
<ControllerSet name="Controllers">
  <objects/>
  <groups/>
</ControllerSet>

<!--XML file (.xml) containing the forces applied to the model as
    ExternalLoads.-->
<external_loads_file> subject01_externalForces.xml </external_loads_file>

<!--Motion (.mot) or storage (.sto) file containing the desired point
    trajectories.-->
<desired_points_file> </desired_points_file>

<!--Motion (.mot) or storage (.sto) file containing the desired kinematic
    trajectories.-->
<desired_kinematics_file> subject01_walk1_RRA_Kinematics_initial_q.sto </desired_kinematics_file>

<!--File containing the tracking tasks. Which coordinates are tracked and
    with what weights are specified here.-->
<task_set_file> gait2354_CMC_Tasks.xml </task_set_file>

<!--File containing the constraints on the controls.-->
<constraints_file> gait2354_CMC_ControlConstraints.xml </constraints_file>

<!--File containing the controls output by RRA. These can be used to place
    constraints on the residuals during CMC.-->
<rra_controls_file> </rra_controls_file>

<!--Low-pass cut-off frequency for filtering the desired kinematics. A
    negative value results in no filtering. The default value is -1.0, so
    no filtering.-->
<lowpass_cutoff_frequency> -10.00000000 </lowpass_cutoff_frequency>

<!--Time window over which the desired actuator forces are achieved.
    Muscles forces cannot change instantaneously, so a finite time window
    must be allowed. The recommended time window for RRA is about 0.001
    sec, and for CMC is about 0.010 sec.-->
<cmc_time_window> 0.01000000 </cmc_time_window>

<!--Flag (true or false) indicating whether or not to use the curvature
    filter. Setting this flag to true can reduce oscillations in the
    computed muscle excitations.-->
<use_curvature_filter> false </use_curvature_filter>

<!--Flag (true or false) indicating whether to use the fast CMC
    optimization target. The fast target requires the desired
    accelerations to be met. The optimizer fails if the acclerations
```

```
        constraints cannot be met, so the fast target can be less robust. The
        regular target does not require the acceleration constraints to be
        met; it meets them as well as it can, but it is slower and less
        accurate.-->
    <use_fast_optimization_target> true </use_fast_optimization_target>

    <!--Preferred optimizer algorithm (currently support "ipopt" or "cfsqp",
        the latter requiring the osimFSQP library.-->
    <optimizer_algorithm> ipopt </optimizer_algorithm>

    <!--Perturbation size used by the optimizer to compute numerical
        derivatives. A value between 1.0e-4 and 1.0e-8 is usually
        approprieate.-->
    <optimizer_derivative_dx> 1.00000000 </optimizer_derivative_dx>

    <!--Convergence criterion for the optimizer. The smaller this value, the
        deeper the convergence. Decreasing this number can improve a solution,
        but will also likely increase computation time.-->
    <optimizer_convergence_criterion> 0.00010000 </optimizer_convergence_criterion>

    <!--Maximum number of iterations for the optimizer.-->
    <optimizer_max_iterations> 200 </optimizer_max_iterations>

    <!--Print level for the optimizer, 0 - 3. 0=no printing, 3=detailed
        printing, 2=in between-->
    <optimizer_print_level> 0 </optimizer_print_level>

    <!--True-false flag indicating whether or not to turn on verbose printing
        for cmc.-->
    <use_verbose_printing> false </use_verbose_printing>

        <!--List of individual Actuators by individual or user-defined group name to be excluded from CMC's
control.-->
         <actuators_to_exclude />

 </CMCTool>

</OpenSimDocument>
```

Specifying the Model

The <model_file> specifies the name of the .osim file used to construct and initialize the model. The <force_set_files> are a list of actuators set file names, each of which may be included as actuators of the model. These actuators are in addition to any actuators specified in the model file that are already part of the model. <replace_force_set>, if set to true, indicates that the actuator sets listed in <force_set_files> should replace any actuators that are already part of the model otherwise actuators are appended.

Specifying Initial and Final Times

The properties <initial_time> and <final_time> specify the time interval over which CMC is to be run. Be aware that CMC uses 0.030 seconds starting from the specified initial time to initialize any muscle states. In addition, to avoid undesirable effects having to do with the processing of the desired kinematics and external loads, it is prudent not to start CMC at the very beginning of the desired kinematics but to allow a buffer. One or two percent of a gait cycle, for example, is usually sufficient.

Specifying Integrator Settings

The <maximum_number_of_integrator_steps> property indicates the maximum number of steps CMC can take during a run of the tool before terminating. Making the <integrator_accuracy> finer (smaller) will decrease the integrator step size.

Specifying Analyses and Results

Any number of available analyses can be added to a CMC run. To get a listing of available analyses, open the CMC tool in the GUI and go the the Analyses pane and click on Add or see the Analyses chapter.

There are several properties associated with the analyses results. <results_directory> specifies the directory where results should be written. <output_precision> specifies how many decimal places should be used when writing results. A value of 20 is sufficient to avoid round-off error when reading results back in during other steps in the workflow. <step_interval> specifies how often to record results during numerical integration. A value of 10 means record results every 10 integration steps.

Specifying Desired Kinematics

The file containing the desired kinematics is specified using the property <desired_kinematics_file>. A low-pass cutoff frequency for filtering the kinematics can be specified using the property <lowpass_cutoff_frequency>. An order 50 Finite Impulse Response Filter (FIR) is currently used. If the desired kinematics have already been filtered, as is normally the case by the time the CMC tool is used, specify a negative cutoff frequency to prevent any filtering at all.

Specifying External Loads

There can be multiple sets of ground reaction forces for different locations on the body. These are specified in the file identified by the <external_loads_file> tags. The external loads .xml file specifies the source file for input forces and to which location and body the forces are to be applied. See Inverse Dynamics for additional details for applying any number of external loads as point forces and/or body torques.

> ⓘ Note: Across all steps in the OpenSim workflow, it is important to use the same settings for specifying the external loads.

Controlling the Optimizer

A number of properties are available for controlling the optimizer. <use_fast_ optimization_target> specifies whether the fast or slow optimization target should be used (How RRA Works). <optimizer_derivative_dx> specifies the perturbation size in the controls for computing numerical derivatives. <optimizer_convergence_criterion> specifies a convergence criterion; the smaller this value, the deeper the convergence. <optimizer_max_iterations> specifies an iteration limit for each time step. <optimizer_print_level> allows diagnostic information to be printed. The larger the number (up to a maximum of 3), the more information that is printed.

Settings for the CMC algorithm

A number of properties are available for controlling the CMC algorithm proper. <cmc_time_window> specifies the time allowed for time-dependent actuators like muscles to change force. If the window is too small, actuator forces will not have enough freedom to generate the desired accelerations. If the window is too large, the control will be too coarse to allow for good tracking (How RRA Works). <use_curvature_filter> specifies whether or not to apply a curvature filter to computed excitations. If control values are oscillating, the curvature filter can used to attenuate these oscillations. To apply the curvature filter, set <use_curvature_filter> to **true**. <adjust_com_to_reduce_residuals> is a property used for residual reduction (Chapter 3). During CMC, this property should be set to **false**.

Specifying Tracking Tasks and Control Constraints

Which coordinates should be tracked and how those coordinates should be tracked are specified in a separate XML file specified by the property <task_set_file>. Constraints on the controls (e.g., muscle excitations) are also specified in a separate XML file using the property <constraints_file>. These files are covered below.

Specifying Actuators to Exclude

In version 3.1 and later, you can tell CMC to ignore specific actuators by adding them to the list of actuators using the property <actuators_to_exclude>. This is useful if you have, for example, a model component with its own controller.

Task Set File

A task set is used to specify which coordinates (e.g., joint angles) in a model should follow a desired kinematic trajectory. It also specifies the relative weighting between coordinates.

Task Sets

Tasks, the class objects used to specify a tracking goal, are kept in sets. Sets are lists of objects. The individual tasks are found between opening and closing xml tags <objects> and </objects>.

Task Types

Although there will be additional task types in the future, such as orientation and point tracking, currently only tracking of generalized coordinates (or joints) is supported. The properties for each joint task are enclosed between opening and closing XML tags <CMC_Joint> and </CMC_Joint>.

Example 2: XML file for the CMC task set file

gait2354_CMC_tasks.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<CMC_TaskSet name="gait2354_CMC">

  <objects>

    <CMC_Joint name="pelvis_tz">
      <on> true </on>
      <active> true false false </active>
      <weight> 1.0e0 </weight>
      <kp> 100.0 </kp>
      <kv> 20.0 </kv>
      <coordinate> pelvis_tz </coordinate>
    </CMC_Joint>

    <CMC_Joint name="hip_flexion_r">
      <on> true </on>
      <active> true false false </active>
      <weight> 1.0e2 </weight>
      <kp> 100.0 </kp>
      <kv> 20.0 </kv>
      <coordinate> hip_flexion_r </coordinate>
    </CMC_Joint>

  </objects>

</CMC_TaskSet>
```

Task Properties

The property <on> is used to turn a task on or off. The <active> property specifies which component of task is active. While other task types might have three components, joint tasks only have one, so the typical setting is **true false false**. Properties <kp> and <kv> specify the position and velocity error gains, respectively. See How RRA Works for an explanation of error gains. The <coordinate> property specifies to which model coordinate a task applies. The value of the property must agree with the name of the model coordinate exactly. Joints can be translational (e.g., pelvis_tz) or rotational (e.g., hip_flexion_r).

Control Constraints File

Control Sets

The properties governing the controls in a simulation are contained in a <ControlSet>. Controls are functions that vary as a function of time. Currently, two types of controls are supported: <ControlConstant> and <ControlLinear>. <ControlConstant> objects are constant for any value of time. <ControlLinear> objects consist of an array of paired time and control values or nodes (i.e., [t1,x1], [t2,x2], … [tn,xn]). These values are linearly interpolated to determine the value of the control at a particular time. <ControlLinear> is the class most commonly used during CMC and is reviewed further below.

Default Objects

Many control objects within a file may have the same values for a number of properties. At the beginning of a control set, there is a defaults section, demarcated by the opening and closing tags <defaults> and </defaults>, that can be used to specify default property values. If a control has the same value as the default control, that property need not be repeated.

Properties of <ControlLinear> Control Type

A number of properties are used to describe the <ControlLinear> control type.

- <is_model_control> indicates whether or not a control belongs to the model. Muscle excitations and actuator controls in general are examples of such controls. For specifying control constraints for CMC, this property should always be **true**.
- <extrapolate> specifies that the value of the control should be extrapolated when the time value is outside the valid time range of a control (e.g., before t1 or after tn).
- <default_min> and <default_max> specify the default minimum and default maximum values, respectively, of a control. For muscles, these are frequently **0.02** and **1.00** as in Example 3, below

- <use_steps> specifies whether or not the value of a control should be linearly interpolated between nodes. If set to **true**, a control is treated as a step function in which the value of a step is determined by the value of the nearest node to the right or, equivalently, later in time. For CMC, it generally works better for muscle excitations to use steps.

Example 3: XML file for the CMC control constraints file

gait2354_CMC_ControlConstraints.xml

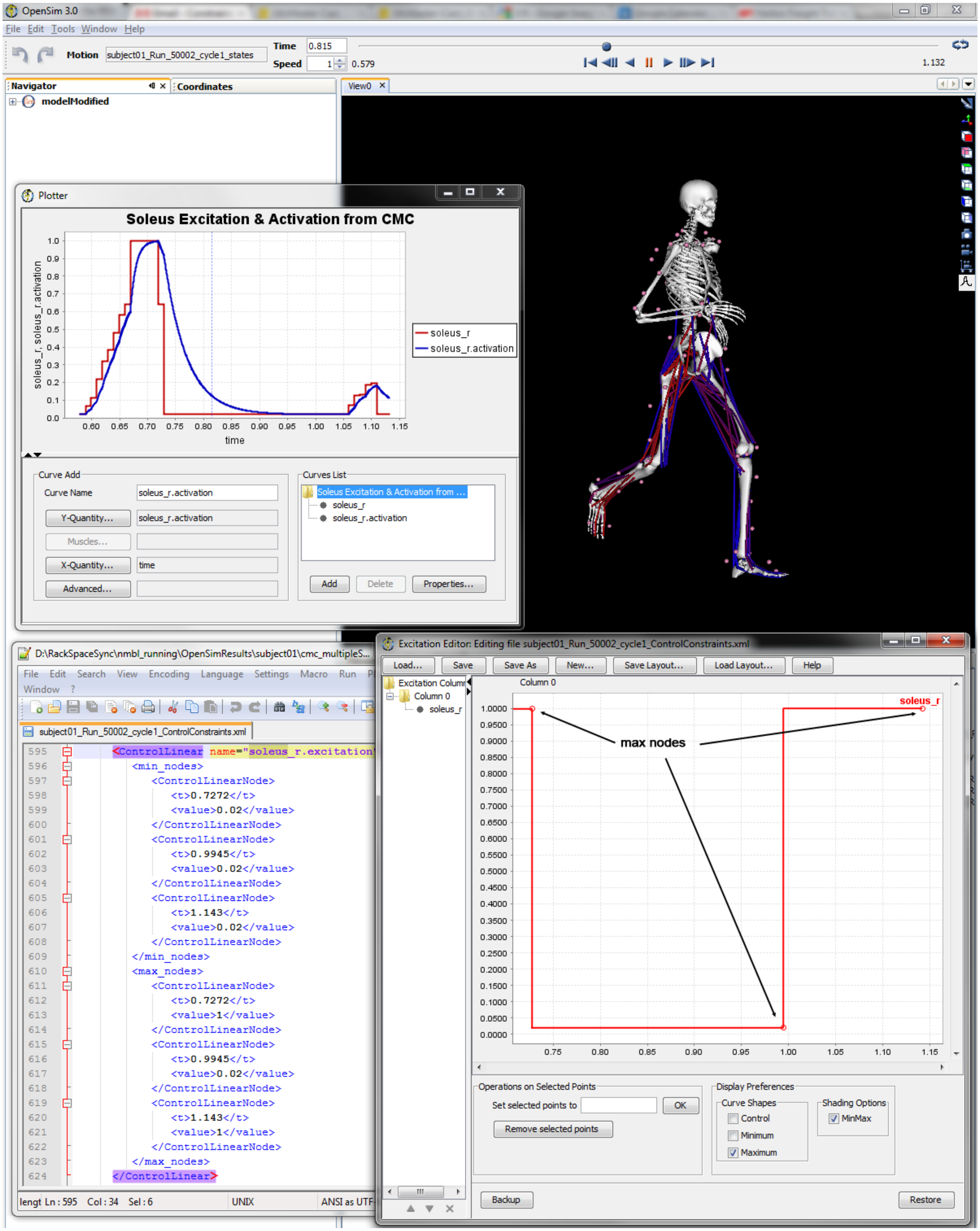
```
<?xml version="1.0" encoding="UTF-8"?>

  <ControlSet name="gait2354_CMC">

    <defaults>

      <ControlLinear name="default">
        <is_model_control> true </is_model_control>
        <extrapolate> true </extrapolate>
        <default_min> 0.02 </default_min>
        <default_max> 1.00 </default_max>
        <use_steps> true </use_steps>
      </ControlLinear>

    </defaults>

    <objects>

      <!--Residual actuators -->
      <ControlLinear name="FX.excitation">
        <default_min> -1.0 </default_min>
        <default_max> 1.0 </default_max>
        <use_steps> false </use_steps>
      </ControlLinear>

      <!-- Reserve actuators -->
      <ControlLinear name="hip_flexion_r_reserve.excitation">
        <default_min> -1000 </default_min>
        <default_max> 1000 </default_max>
      </ControlLinear>

      <!-- Muscles -->
      <ControlLinear name="glut_med1_r.excitation" />
      <ControlLinear name="glut_med2_r.excitation" />
      <ControlLinear name="glut_med3_r.excitation" />
      <ControlLinear name="bifemlh_r.excitation" />
      <ControlLinear name="bifemsh_r.excitation" />

  </objects>

</ControlSet>
```

Example 4: Using control constraints file to constrain CMC excitations to match EMG data

When using CMC, sometimes the optimizer will choose to activate a muscle during a part of the movement when it should not be activated. For example, based on EMG recordings during running, we know that soleus should not have much activity during swing. But sometimes CMC will choose to activate the muscle since it is not informed about the EMG data. To get a better match with experimental EMG, we can constrain the activation CMC chooses for soleus during a specific time period of the movement.

Below is an example of a constraint, where the right soleus is specified to be off between 0.7272 and 0.9945s.

The control constraints are displayed in the Excitation Editor GUI. Lines have been added to illustrate how the nodes are connected. The value associated with a node is applied **before** the specified time. In other words, when interpreting a control or its constraint (min or max value) it is the value from the future node at t(i+1) to the present node at t(i) that is applied.


OpenSim 3.0
File Edit Tools Window Help
Motion subject01_Run_50002_cycle1_states
Time 0.815
Speed 1
0.579
1.132
Navigator
modelModified
Coordinates
View0
Plotter
Soleus Excitation & Activation from CMC
soleus_r, soleus_r.activation
time
soleus_r
soleus_r.activation
Curve Add
Curve Name soleus_r.activation
Y-Quantity... soleus_r.activation
Muscles...
X-Quantity... time
Advanced...
Curves List
Soleus Excitation & Activation from ...
soleus_r
soleus_r.activation
Add
Delete
Properties...
D:\RackSpaceSync\nmbl_running\OpenSimResults\subject01\cmc_multipleS...
File Edit Search View Encoding Language Settings Macro Run
Window ?
subject01_Run_50002_cycle1_ControlConstraints.xml
lengt Ln : 595 Col : 34 Sel : 6 UNIX ANSI as UTF
Excitation Editor: Editing file subject01_Run_50002_cycle1_ControlConstraints.xml
Load... Save Save As New... Save Layout... Load Layout... Help
Excitation Column
Column 0
soleus_r
max nodes
Operations on Selected Points
Set selected points to OK
Remove selected points
Display Preferences
Curve Shapes
☐ Control
☐ Minimum
☑ Maximum
Shading Options
☑ MinMax
Backup
Restore

```
<ControlLinear name="soleus_r.excitation
    <min_nodes>
        <ControlLinearNode>
            <t>0.7272</t>
            <value>0.02</value>
        </ControlLinearNode>
        <ControlLinearNode>
            <t>0.9945</t>
            <value>0.02</value>
        </ControlLinearNode>
        <ControlLinearNode>
            <t>1.143</t>
            <value>0.02</value>
        </ControlLinearNode>
    </min_nodes>
    <max_nodes>
        <ControlLinearNode>
            <t>0.7272</t>
            <value>1</value>
        </ControlLinearNode>
        <ControlLinearNode>
            <t>0.9945</t>
            <value>0.02</value>
        </ControlLinearNode>
        <ControlLinearNode>
            <t>1.143</t>
            <value>1</value>
        </ControlLinearNode>
    </max_nodes>
</ControlLinear>
```

Control Constraints XML to Constraint Soleus

```
<ControlLinear name="soleus_r.excitation">
     <min_nodes>
          <ControlLinearNode>
               <t>0.7272</t>
               <value>0.02</value>
          </ControlLinearNode>
          <ControlLinearNode>
               <t>0.9945</t>
               <value>0.02</value>
          </ControlLinearNode>
          <ControlLinearNode>
               <t>1.143</t>
               <value>0.02</value>
          </ControlLinearNode>
     </min_nodes>
     <max_nodes>
          <ControlLinearNode>
               <t>0.7272</t>
               <value>1</value>
          </ControlLinearNode>
          <ControlLinearNode>
               <t>0.9945</t>
               <value>0.02</value>
          </ControlLinearNode>
          <ControlLinearNode>
               <t>1.143</t>
               <value>1</value>
          </ControlLinearNode>
     </max_nodes>
</ControlLinear>
```

Residual Actuators

Residual actuators are actuators that act directly between the model and the ground. These are used to control the global position and orientation of a model. If the residuals for a simulation have been reduced (see Residual Reduction Algorithm), the loads applied by these actuators should be relatively small. However, unlike muscles, the residual actuators can apply negative as well as positive forces. The minimum and maximum control values here, therefore, differ from the default values set at the top of the example file (0.02 and 1.0) and must be specified explicitly (-1.0 and 1.0).

Reserve Actuators

Reserve actuators are actuators that can make up for insufficient muscle strength during a simulation. However, they should only apply significant loads (e.g., forces above 1 N or 1 Nm) when necessary. To penalize the use of reserve actuators, the minimum and maximum control values are allowed to be very large (-1000 and 1000 in Example 3 above). Forces are specified in newtons and torques in newton-meters.

Muscles

When a control has the same property values as that specified in the defaults section at the top of the control constraints file (Example 3), it is only necessary to have a one-line entry that specifies the name of the control.

Next: Analyses

Previous: How to Use the CMC Tool

Home: Computed Muscle Control