

New Features in OpenSim 3.2

In [OpenSim 3.2](#), we've focused on improving the OpenSim scripting interface, accessible through the Graphical User Interface (GUI), Matlab, and now Python. We've also added new visualization capabilities and usability improvements in the OpenSim application. As with every release, we've also implemented performance improvements and bug fixes. Improvements and additions include:

- [GUI Usability and Visualization](#)
- [Scripting for Matlab, the GUI, and now Python](#)
- [Performance and Accuracy Improvements](#)
- [OpenSim Tools](#)
- [API and Advanced User Improvements](#)
- [Examples](#)
- [Bug Fixes](#)

GUI Usability and Visualization

- Auto-updates for GUI improvements and bug fixes. See [Updating OpenSim](#)
- Color muscles from an input file, for example static optimization results. See [Associating Data with a Motion](#).
- Visualize the structure of OpenSim models with the model topology view. See [Windows](#).
- Keyboard shortcuts for common actions. See [Menus](#).
- Easy selection of model components via double click across all panels and views in the GUI.
- Auto-population (with the old name) when renaming models, motions, and other objects in the GUI

Scripting for Matlab, the GUI, and now Python

Scripting through Matlab, the GUI, and in Python allows users to call the full set of OpenSim modeling and simulation libraries (the API) without learning C++. Scripting also allows users to create batch scripts to streamline their workflows. We improved the OpenSim scripting interface for OpenSim 3.2. Enhancements include:

- A Matlab install file to streamline the setup process
- Improved documentation and getting started guide. See [Scripting and Development](#)
- New GUI scripting calls to plot the sum of curves and get a handle to the current motion
- Access to the InverseKinematicsSolver through scripting
- Access to MarkersReference, CoordinateReference, and SetMarkerWeights
- Utility methods to append Vec and Vec3 to ArrayDouble objects in scripting
- Utility to create a marker data object from a storage file
- Clean up of the interface by removal of many unusable utility classes
- Scripting in Python (for advanced users)

Learn more about [Scripting](#) in the User's Guide.

Performance and Accuracy Improvements

- Correction to moment arm calculations for models with moving path/muscle points
- Fix to memory leaks (e.g., ExternalForces/ExternalLoads and printing of objects to XML)
- Fixed bug in Hunt & Crossley force: large dissipation coefficient could cause a negative force which then produced an unstable friction force that prevented time integration from progressing.

OpenSim Tools

- Eliminated nested files for IK (along with the bug fix, described below). Users should now use one setup file for IK. Examples have been updated accordingly.
- Improved CMC error message. We updated the message to reflect the advice we give to users to make sure that they have all dofs actuated, that there are no constraints (e.g. locks) on coordinates being tracked and that there are no constraints on the controls to reserve and residual actuators.
- Exposed stepInterval for Static Optimization in the GUI.
- Added the ability to set the output filename for IK.

API and Advanced User Improvements

- Exposed stretch for ClutchedPathSpring. Added initial_stretch to the properties of the ClutchedPathSpring so that it can be used to initialize the stretch state of the path spring prior to simulation.
- Added support Visual Studio 2013
- Added a new test case for metabolics
- Full conversion to replace simtklapack
- Modular muscle activation dynamics and other muscle class cleanup
- Build/dependency cleanup
- Unified the implementation of connectToModel in the base Joint class and removed overridden methods in the individual concrete classes.
- Conversion to Simbody 3.3
- Create buildinfo.txt file under installation/sdk to include Product Version, Compiler and Platform used to make the build.
- Clean up of stale classes

Examples

We've added to the library of OpenSim examples and tutorials. New and updated materials include:

- Scripts to run the OpenSim tutorials in the GUI
- Tutorial 1 was converted to use the gait2392 model
- Clean up of models folder to remove BothLegs and SepLegs

Bug Fixes

- Added option to create a series of still frames (instead of .avi file) as a workaround to bug creating movies on the 64-bit build of OpenSim.
- Fixed bug respecting coordinate inputs to Scale
- Fixed error in Analysis updY, affecting plotter
- Fixed bug in estimateMuscleFiberState()
- Fix to handle .MOT files (i.e., capitalized file extension)
- Fix to Gimbal joint bug when scaling. It was overriding the scale method but it was implemented to do nothing, which resulted in a scaling of Gimbal joint that had no change in the joint dimensions.
- Fix bug where running a script thru the menu causes the scripting shell to stop responding to input
- Fixed bug in constructor for Umberger2010MuscleMetabolicsProbe_MetabolicMuscleParameter, which is called by Umberger2010MuscleMetabolicsProbe::addMuscle(). isnan() was checking the existing provided_muscle_mass, not the input (which is NaN by default). Specifying a mass when calling addMuscle() had no effect.
- Fixed bugs in constructors for Bhargava2004MuscleMetabolicsProbe_MetabolicMuscleParameter. Specifying a mass when calling addMuscle() had no effect. Analogous to fixes made in Umberger probe (r8543).
- Fixed bug in Bhargava2004MuscleMetabolicsProbe. Unnormalized rates (in Watts) were being multiplied by muscle mass. Bhargava2004 and Umberger2010 probes now report the same mechanical work rate, as expected.
- Fixed bug in Thelen2003Muscle::printCurveToCSVFile(), as found by Gabriel Baud-Bovy (bug #1956). Second input argument to calcfvInv() and calcDdlceDaFalFv() should be fal, not a*fal.
- Fixed bugs for cancelling tools in the GUI
- Fixed bug in TorqueActuator which was not applying equal and opposite torque on respective bodies. The second body was being ignored by default (so effectively ground). Also fixed a mistake in the resulting actuator powers so that it is dependent on the relative velocity between bodies and not just bodyA. The only use case was residual actuators with respect to ground so problematic scenarios were never exercised by any of our models, examples or test cases. Therefore, a test case was added to testActuators.
- RigidTendonMuscle: The passiveFiberForceMultiplier (e.g. the passive force length curve) should never be a negative value. We now apply a clamp in case the user-supplied spline goes negative.
- Fixed bug in Millard2012EquilibriumMuscle. In estimateMuscleFiberState(), the path velocity is shared between the fiber and tendon based on their relative stiffnesses, but the algorithm can produce nonzero velocities even if the path velocity is zero. Now returns static solution when expected.
- Fix writing of deprecated boolArray properties so that they are read back correctly. Fixes bug reported during workshop with CMCTasks.
- Enable re-running IK from GUI after interrupting.