How to Build a Dynamic OpenSim Model Based on the SDFast Dynamics Engine

OpenSim supports two dynamics engines: Simbody and SDFast. Simbody is the open-source dynamics engine being developed as part of SimTK (https://simtk.org/home/simbody). Currently, Simbody does not support general, user-defined constraints and, therefore, may not be suitable for modeling certain kinds of joints. SDFast is a commercial, widely-used dynamics engine offered by Parametric Technology Corporation (http://www.sdfast.com). It does support general, user-defined constraints and is the dynamics engine used by the SIMM Dynamic Pipeline (http://www.musculographics.com).

SDFast generates equations of motion for a model in symbolic form. Specifically, based on a system description file (e.g., gait23.sd), it generates highly efficient source code (e.g., gait23_dyn.c) that solves the equations of motion. To use this source code as the basis of an OpenSim model, this source code must be complied into a dynamically linked library or DLL.

A fully-functional SDFast-based OpenSim model requires two files: 1) a model DLL (described just above) and an OpenSim model file. An OpenSim model file, which has the file extension .osim, contains the complete specification of all the objects and properties that constitute a model, including, for example, the body segments, generalized coordinates, muscles, wrap objects, etc.

The general procedure for building an SDFast-based model is to

A. Convert an existing SIMM model (your_model.jnt, your_model.msl) into a kinematic OpenSim model (your_model.osim) using the **simmToOpenSim** utility.

B. Convert the kinematic model into an SDFast-based model using the **makeSDFModel** utility. This produces an SDFast-based OpenSim model file (your_model_sdfast.osim) and a system description file (e.g., your_model.sd).

C. Run SDFast on the system description file to generate the source code for the dynamic model (e.g., your_model_dyn.c, your_model_sar.c).

D. Compile the source code for the dynamic model into a DLL (your_model.dll).

Note that you don't always need to follow this procedure from the beginning. The entry point for building a dynamic model may be A, B, C, or D, depending on what is needed. For example, if you already have a system description file for an SDFast model (your_model.sd) and the sdfast-based OpenSim model file (your_model_sdfast.osim), you would start with C. Currently, the only way to generate an OpenSim model file is either to convert an existing SIMM model or create one manually using a text editor.

The specific steps necessary to build a dynamic model are detailed below.

Pre-requisites

1. A working installation of SDFast. www.sdfast.com
2. A C++ compiler. Microsoft Visual Studio 2003 (7.1) is recommended, but other compilers may work as well. Visual Studio 2008 Express Edition can be obtained for free. msdn2.microsoft.com/en-us/express/default.aspx
3. CMake 2.4.6 (or higher). www.cmake.org
4. The OpenSim source code. At the present time, you must be a member of the OpenSim project to download the OpenSim source code. The source code is checked out from the URL https://simtk.org/home/opensim/trunk using Subversion (www.subversion.org).

Preparation

1. If you have not already done so, create a folder for containing all the source files related to the model you are creating. You may pick any name for this folder you wish. Throughout the rest of this document, the name of this folder will be assumed to be **MySDFastModel**. *Do not create this folder inside the OpenSim source code tree.* In this documentation, by way of example, the name of the model is assumed to be **mysdfastmodel**.

A. Convert an existing SIMM model into a kinematic OpenSim model.

1. Copy the appropriate SIMM joint and muscle files into the folder you just created. To convert any bones files into OpenSim-compatible geometry files, copy all the bones files associated with a model into a local directory called bones. So, for example, you should have a folder structure that looks something like this:

```
        MySDFastModel/
                bones/
                        femur.asc
                        patella.asc
                        …
                mysdfastmodel _simm.jnt
                mysdfastmodel _simm.msl
```

2. Using a command prompt (or equivalent shell program, such as cygwin), execute the following command:

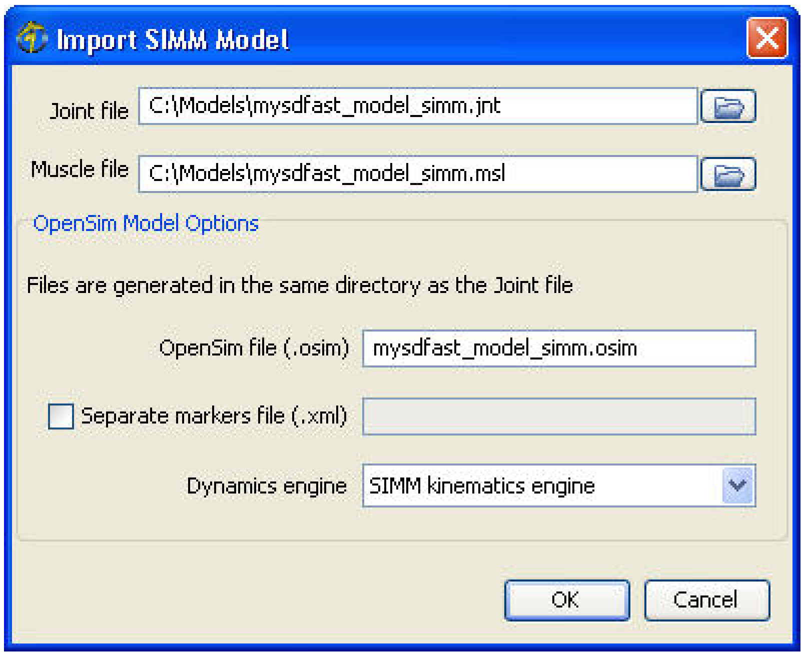
   prompt> simmToOpenSim -j *mysdfastmodel _simm.jnt* -m *mysdfastmodel_simm.msl* -x mysdfastmodel _simm.osim -g Geometry

   This should create an OpenSim model file based on the SIMM kinematics engine called **mysdfastmodel _simm.osim**, as well as a number of geometry files for the bones in a folder called **Geometry**.

   An alternate way to do this is to invoke the following command from the GUI:

   File->Import SIMM Model

   And browse for the joint, muscle file, pick “SIMM kinematics engine” from the dropdown and enter “mysdfastmodel_simm.osim” for OpenSim file (.osim). If you do this step in the GUI you’ll be able to edit the model in the GUI as well.



Joint file: C:\Models\mysdfast_model_simm.jnt

Muscle file: C:\Models\mysdfast_model_simm.msl

OpenSim Model Options

Files are generated in the same directory as the Joint file

OpenSim file (.osim): mysdfast_model_simm.osim

☐ Separate markers file (.xml)

Dynamics engine: SIMM kinematics engine

B. Convert the kinematic model into an SDFast-based model.

1. Using a command prompt (or equivalent shell program, such as cygwin), execute the following command:

   prompt> makeSDFastModel -InputModel *mysdfastmodel_simm.osim* -SystemDescription *mysdfast_model.sd* -ModelLibrary *mysdfastmodel* -OutputModel *mysdfastmodel_sdfast.osim*

   This should create an OpenSim model file based on an SDFast dynamics engine called **mysdfastmodel_sdfast.osim** as well as a system description file called **mysdfastmodel.sd** that will be used by SDFast to generate the equations of motion. Note: Do not use the -FF option to generate a forward dynamics source file. If you do, CMake (see D.3 below) will add it to the project and the DLL will not compile properly.

C. Run SDFast to generate the source code for the dynamic model.

1. Using a command prompt (or equivalent shell program, such as cygwin), execute the following command to generate the dynamics files for your model:

   prompt> sdfast

   When you are prompted for the system description file, specify the system description file generated in step B1 above (e.g., **mysdfast_model.sd**). You may accept the default names suggested by SDFast for the generated dynamics files or change them to something of your choice. You must, however, keep the extensions of these files .c. This step should create two new files: **mysdfastmodel_dyn.c** and **mysdfastmodel_sar.c**. The file **mysdfastmodel_dyn.c** contains the equations of motion; **mysdfastmodel_sar.c** contains simple analysis routines needed by SDFast for some SDFast models.

2. If you do not already have an **sdlib.c** file generated on your computer, use SDFast to generate one now by executing the following command:

   prompt> sdfast –g l

   This source file contains standard library routines used by SDFast. It must be generated using the same installation of SDFast as the dynamics files generated in step 1 or SDFast will exit with an error. SDFast inserts numeric keys in the generated files to ensure that the files are not used on computers for which they were not licensed.

   3. Copy the files generated in steps 1 & 2 to the directory **MySDFastModel** that you created in the "preparation" step above.

D. <u>Compile the source code for the dynamic model into a DLL.</u>

1. If you have not already done so, place a copy of the file **CMakeLists_sdfastmodel.txt** into the **MySDFastModel** folder and rename it to **CMakeLists.txt**. This file is located in the OpenSim source code tree under **OpenSim/Models/CMakeLists_sdfast.txt**.
2. Your folder structure should look something like

```
MySDFastModel/
        bones/
        Geometry/
        CMakeLists_sdfastmodel.txt (renamed to CMakeLists.txt)
        mysdfastmodel_simm.jnt
        mysdfastmodel_simm.msl
        mysdfastmodel_simm.osim
        mysdfastmodel_sdfast.osim
        mysdfastmodel_dyn.c
        mysdfastmodel_sar.c
        sdlib.c
```

   Now change the name of the project on the first line of your **CMakeLists.txt** to the name on the of the dll you want for your model.Use the same name you entered for the **ModelLibrary** in step B above.

3. Run CMake to generate a build file for your particular compiler. You will need to specify the following information:
   i. "Where is the source code"

   This is the location of the source code for your models.

   <path to MySDFastModel>

   ii. "Where to build the binaries"

   This is the folder in which to put the generated files.

   <path to MySDFastModel>\Build

   All files generated during the build process will be generated beneath this folder, including the DLL for your model.

iii. EXECUTABLE_OUTPUT_PATH

Leave this field blank.

iv. LIBRARY_OUTPUT_PATH

Leave this field blank.

v. OpenSim_LIB_DIR

This is the location of the prebuilt OpenSim libraries. These libraries are needed so that the generated make/solution file knows where to locate the libraries that the model depends on.

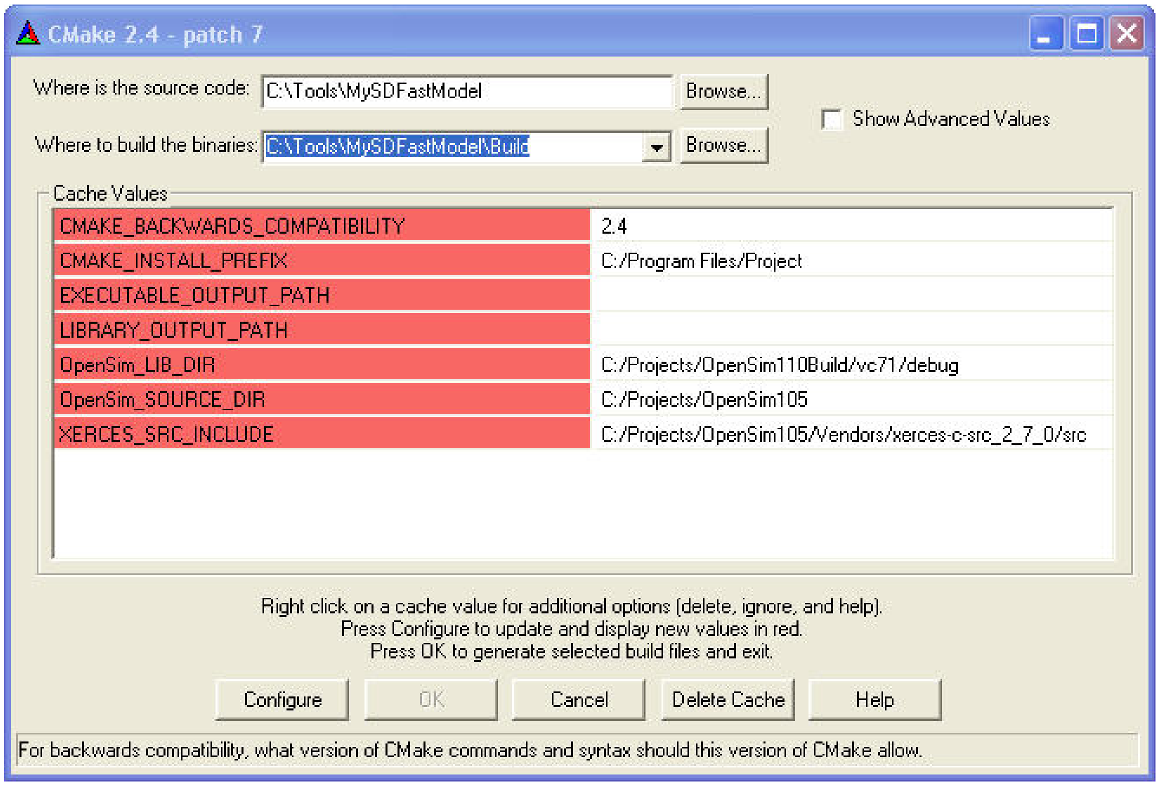
vi. OpenSim_SOURCE_DIR

This is the location of the OpenSim top level source code folder.

vii. XERCES_SRC_INCLUDE

This is the location of the directory containing the xerces library headers.

If you are using Visual Studio 2003, this step should have generated a Visual Studio solution file (**.sln**) in your **MySDFastModel/Build** folder.



Where is the source code: C:\Tools\MySDFastModel

Where to build the binaries: C:\Tools\MySDFastModel\Build

Cache Values

CMAKE_BACKWARDS_COMPATIBILITY	2.4
CMAKE_INSTALL_PREFIX	C:/Program Files/Project
EXECUTABLE_OUTPUT_PATH	
LIBRARY_OUTPUT_PATH	
OpenSim_LIB_DIR	C:/Projects/OpenSim110Build/vc71/debug
OpenSim_SOURCE_DIR	C:/Projects/OpenSim105
XERCES_SRC_INCLUDE	C:/Projects/OpenSim105/Vendors/xerces-c-src_2_7_0/src

Right click on a cache value for additional options (delete, ignore, and help).
Press Configure to update and display new values in red.
Press OK to generate selected build files and exit.

For backwards compatibility, what version of CMake commands and syntax should this version of CMake allow.

4. Build the DLL/shared library for your model. If you are using Visual Studio, open the Visual Studio Solution file generated in the previous step and select **Build All** from the **Build** menu. This should generate a **.lib** file a **.dll** file for your model. These should be located in **MySDFastModel/Build/vc71**. You may generate both debug and release

versions of these libraries. The debug version will be located in a folder called **debug** and have a **_d** following the name of your model. The release version will be located in a folder called **release** and not have an **_d** following the name. On other platforms debug and release libraries may not have different names but simply exist in different directories.

5. Copy the generated libraries to a directory on your path. Your path is the list of directories that the operating system searches for executables and DLLs. Alternatively, add the location of the created libraries to your path. Be aware of the order of directories in your path, as the first DLL found on your path will be the one loaded.