A short introduction to soft tissue simulation

Stefan Suwelack

April 27, 2015

Contents

Part I

Theory

Part II

Tutorial

Chapter 1

Getting started

This chapter describes how to work with virtualized environments that contain pre-configured simulation packages. These environments are based on Docker containers[1] and can either be deployed locally or be run as a server instance.

1.1 Running containers locally

In this section, we describe how to run a pre-configured deployment of the Simulation Open Framework Architecture (SOFA) simulation package.

1. Install the Docker framework for your Linux distribution. Further details can be found on the Docker homepage (e.g. at `http://docs.docker.com/installation/ubuntulinux/` for Ubuntu).

2. Optional: Add your user to the docker group. Otherwise you have to use sudo in front of the docker commands.

3. Pull the SOFA container from the Docker Hub:

   ```
   $ docker pull ssuwelack/msml_sofa
   ```

4. We now start the Docker container as a daemon and bind it's SSH port to port 22000 of the localhost.

   ```
   $ docker run -d -p 127.0.0.1:22000:22 --name
       msml_sofa ssuwelack/msml_sofa
   ```

[1] `www.docker.com`

5. We can use the pre-configured user msml (password: msml) to connect to the running container using SSH:
   ```
   $ ssh –XC msml@localhost –p 22000
   ```
6. The sofa runtime is located at /opt/sofa/bin. All data is stored in the home directory of the msml user.
7. The container can be stopped by executing
   ```
   $ docker stop msml_sofa
   ```
   on the host machine. It can be re-started using the command
   ```
   $ docker start msml_sofa
   ```
   Stopped containers can be deleted through
   ```
   $ docker rm msml_sofa
   ```

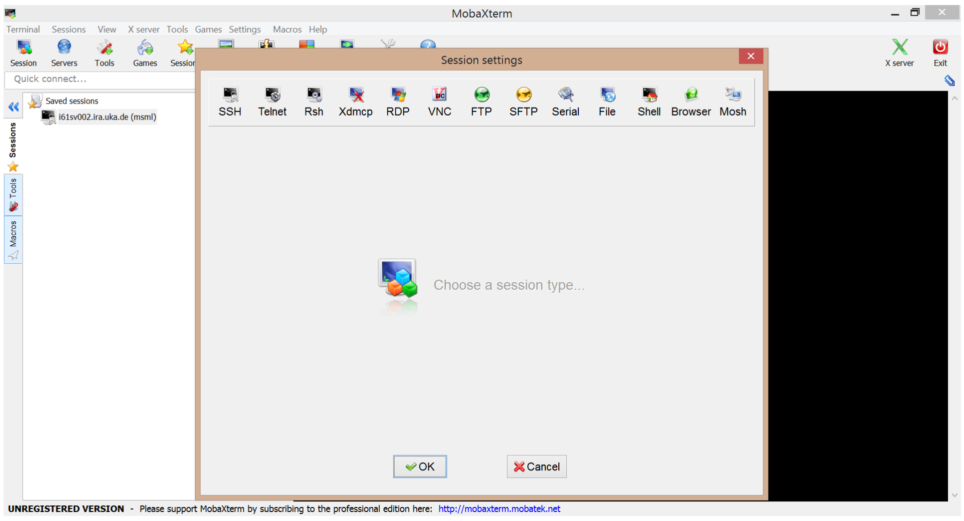
1.2 Connecting to a server instance

In this section, we describe how to run SOFA on a server instance. The server i61sv002.ira.uka.de exposes 25 docker container that run SOFA from port 22001 to 22025. In order to access these containers from a Linux machine inside the the HIS network, simply run

```
$ ssh –XC msml@i61sv002.ira.uka.de –p 22010
```

In the following we describe how the containers can be accessed from any Windows machine.

1. Install the MobaXterm framework which you can download on its homepage (http://mobaxterm.mobatek.net/download-home-edition.html).
2. Open MobaXterm and start a new session (cf. figure 1.1).
3. Configure your SSH session (cf. figure 1.2):
   - server name: i61sv002.ira.uka.de
   - user: msml (password: cmsml)
   - port: choose one from 22001 to 22025
   - As these ports are only accessible inside the HIS network, a jumphost is needed (*Advanced SSH settings*):
     server: i61p24.ira.uka.de
     user/password: use your HIS account


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions
X server Exit
Quick connect...
Saved sessions
i61sv002.ira.uka.de (msml)
Sessions Tools Macros
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh
Choose a session type...
OK
Cancel

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

Figure 1.1: Screenshot of MobaXterm after starting a new session.

1.3 Own Installation

If you need better OpenGL support or more speed, you should install SOFA and MSML by yourself. This process takes a couple of time for installing several dependencies and compiling. You should calculate with two or three hours, depending on your system performance.

1.3.1 SOFA Installation

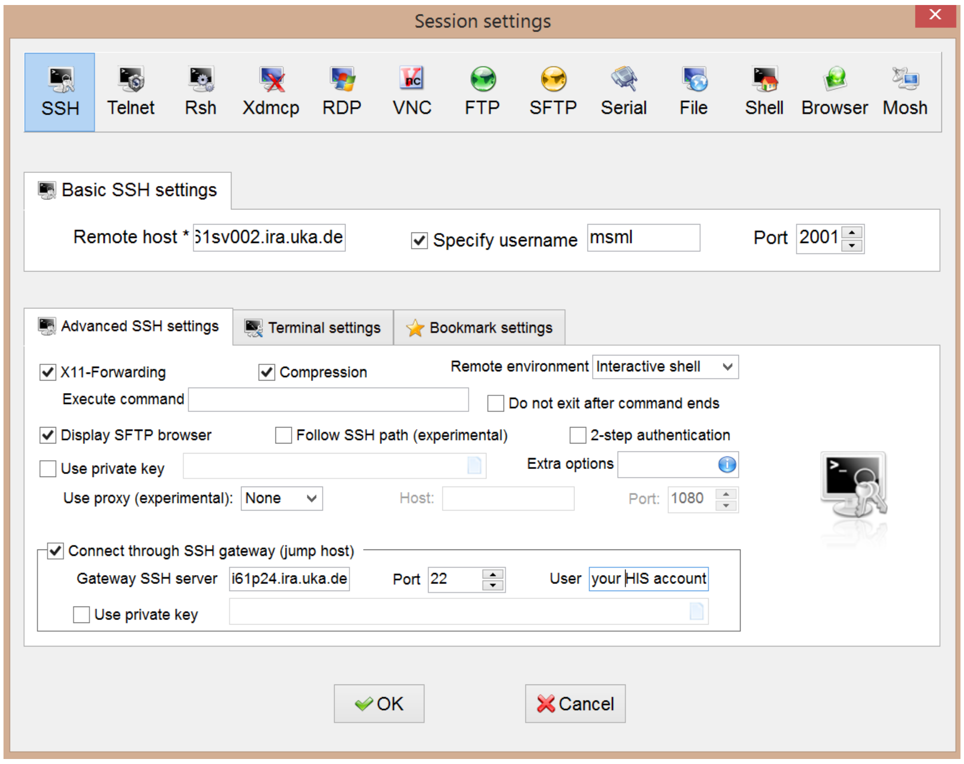
Start by getting the latest SOFA version[2].

```
git clone --depth 1 git://scm.gforge.inria.fr/sofa/
    sofa.git
```

For the compilation you need to install following packages (Ubuntu):

```
sudo apt-get install cmake cmake-qt-gui cmake-curses-
    gui ccache \
                    build-essential libqt4-dev
                       libglew-dev\
                    freeglut3-dev libpng-dev zlib1g-
                       dev\
                    python2.7-hdev libxml2-dev
                       libcgal-dev libblas-dev\
```

[2]More detail: `https://wiki.sofa-framework.org/wiki/Getting_Started`


Session settings

SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh

Basic SSH settings

Remote host * 51sv002.ira.uka.de ☑ Specify username msml Port 2001

Advanced SSH settings | Terminal settings | Bookmark settings

☑ X11-Forwarding ☑ Compression Remote environment Interactive shell

Execute command ☐ Do not exit after command ends

☑ Display SFTP browser ☐ Follow SSH path (experimental) ☐ 2-step authentication

☐ Use private key Extra options

Use proxy (experimental): None Host: Port: 1080

☑ Connect through SSH gateway (jump host)

Gateway SSH server i61p24.ira.uka.de Port 22 User your HIS account

☐ Use private key

OK Cancel

Figure 1.2: Screenshot of MobaXterm after starting a new session.

```
                    liblapack-dev libsuitesparse-dev
                      \
                    libboost-all-dev libassimp-dev
```

Create a new folder, go into it, and trigger CMake.

```
mkdir sofa-build
cd sofa-build
cmake ../sofa
cmake ../sofa
make -j 8
```

1.3.2 MSML

MSML consists of two parts: C++ functionality and a Python core. First, check out the MSML repository:

```
git clone --depth 1 https://github.com/
    CognitionGuidedSurgery/msml.git
```

For C++ we need several libraries (Ubuntu):

```
sudo apt-get install libtet1.5-dev libcgal-dev libvtk6
   -dev \
                     libxml2-dev  \
                     libboost-filesystem-dev libboost-
                        python-dev \
                     libboost-program-options-dev
                        libboost-graph-dev \
                                          libboost-
                        iostreams-dev \
                     python-vtk6 swig python-pip
```

The CMake build process takes care about everything else.

```
  mkdir msml-build
  cd msml-build
  cmake ../msml/operators
  make -j 8
```

The Python dependencies are installed via pip:

```
pip install -r msml/requirements.txt
```