Generating the IKFast CPP file for the IRB 6640 robot

After getting the link info, we can start generating the IK solver CPP file for handling the IK of this robot.

Use the following command to generate the IK solver for the IRB 6640 robot:

```
$ python `openrave-config --python-dir`/openravepy/_openravepy_/ikfast.py --robot=irb6640.dae --iktype=transform6d --baselink=1 --eelink=8 --savefile=output_ikfast61.cpp
```

The preceding command generates a CPP file called `output_ikfast61.cpp` in which the IK type is `transform6d`, the position of the `baselink` is `1`, and the end effector link is `8`. We need to mention the robot DAE file as the robot argument.

We can test this file using the following procedure:

1. Download the IKFast demo code file from http://kaist-ros-pkg.googlecode.com/svn/trunk/arm_kinematics_tools/src/ikfastdemo/ikfastdemo.cpp.
2. Also, copy `IKFast.h` to the current folder. This file is present in the cloned file of OpenRave. We will get this header from `openrave/python`.
3. After getting `output_ikfast61.cpp`, `ikfastdemo.cpp`, and `ikfast.h` on the same folder, we need to edit `ikfastdemo.cpp` and change the following portion. Here, we are commenting a header, and instead of that, we add the CPP file that we have generated, that is `output_ikfast61.cpp`.

   ```
   #define IK_VERSION 61
   #include "output_ikfast61.cpp"
   //#include "ikfast61.Transform6D.0_1_2_3_4_5.cpp"
   ```

4. Compile the edited file and check whether you are getting any errors. Here is the command to compile and execute this code:

   ```
   $ g++ ikfastdemo.cpp -lstdc++ -llapack -o compute -lrt
   $ ./compute
   ```

If the demo is working, we can go to the next step. Now, we have successfully created the IK solver CPP file; the next step is to create a MoveIt! IK Fast plugin using this source code.