# The explanation of UR\_Kinematics package

# 1 [Task Summary]

There is a ur\_kinematics package created specifically for UR series robot to solve the kinematics problem. The ur\_kinematics package encapsulates a moveit\_plugin so we can easily choose and use it through the moveit\_setup\_assistant. Here we are going to explain the package as much as I can.

# 2 [Process]

**Step 1:** Let us take a look at the files in this package.



Through the cd command, we will see 3 .cpp files in the src folder.

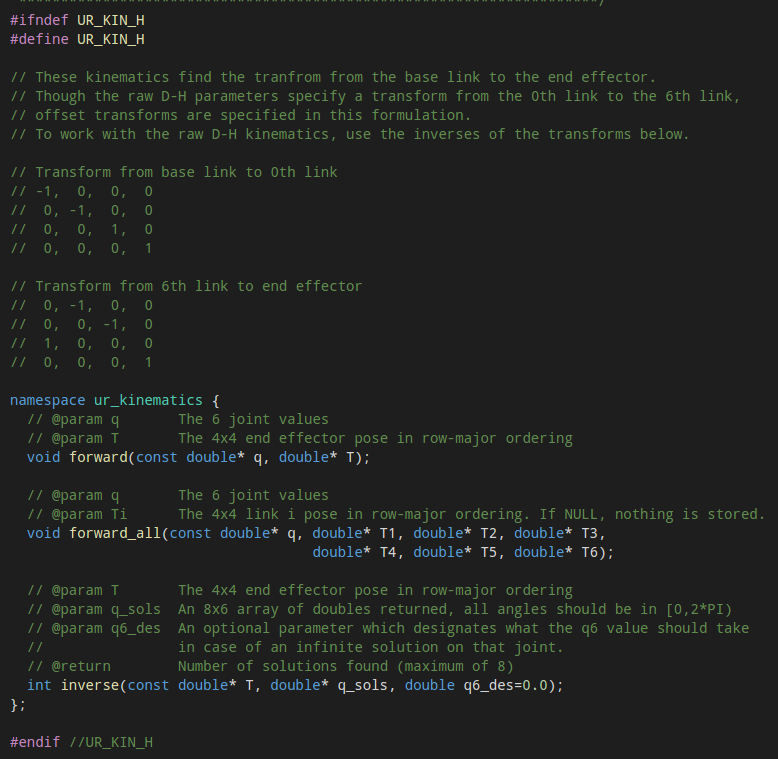
**ur\_kin.cpp** contains the definition of inverse/forward kinematics.

**ur\_kin\_py.cpp** encapsulates the c++ function and turn it into a python interface.

**ur\_moveit\_plugin.cpp** encapsulates the inverse/forward function we created in ur\_kin.cpp and turn it into a moveit\_plugin so we can easily choose it in moveit\_setup\_assistant.

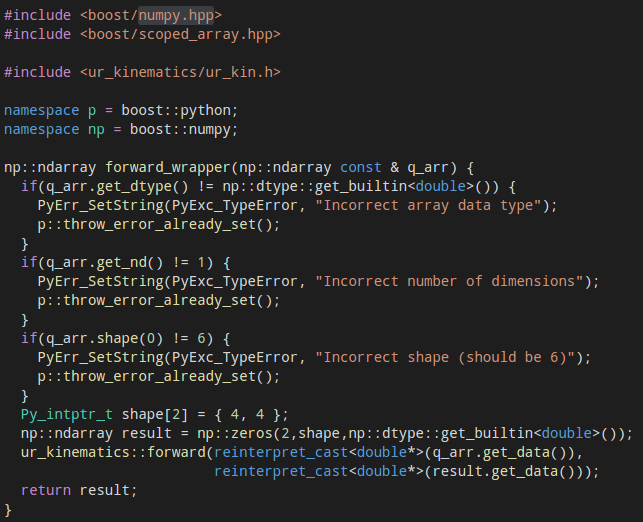
**Step 2:** ur\_kin.cpp and ur\_kin.h.

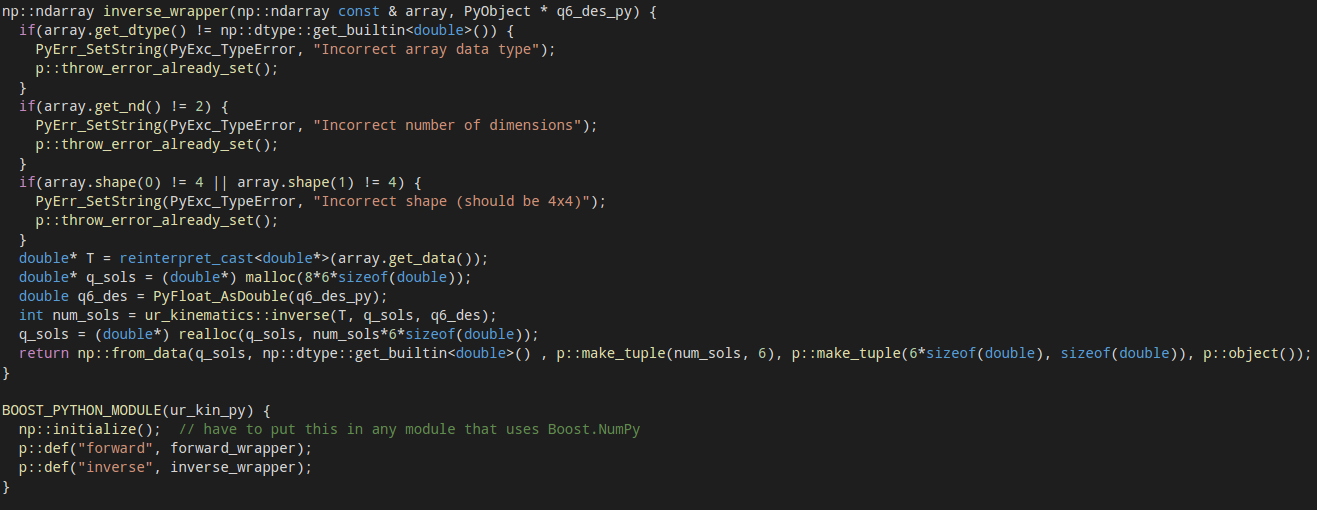
We use IDE to open the ur\_kin.h in the **include** sub\_directory.



Here we can explicitly see all the function’s definition and parameters type.

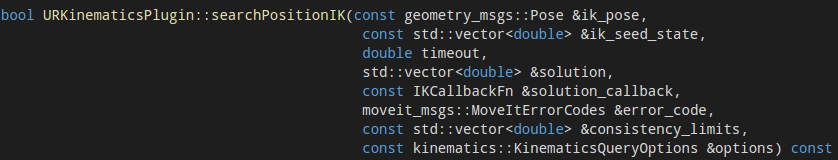
**Step 3:** ur\_kin\_py.cpp



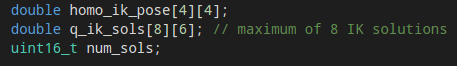


**Step 4:** ur\_moveit\_plugin.cpp

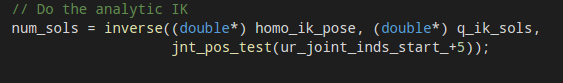
There are too many functions to talk about in this file. I only pick the most important part up -- the method that picks up the best solution.



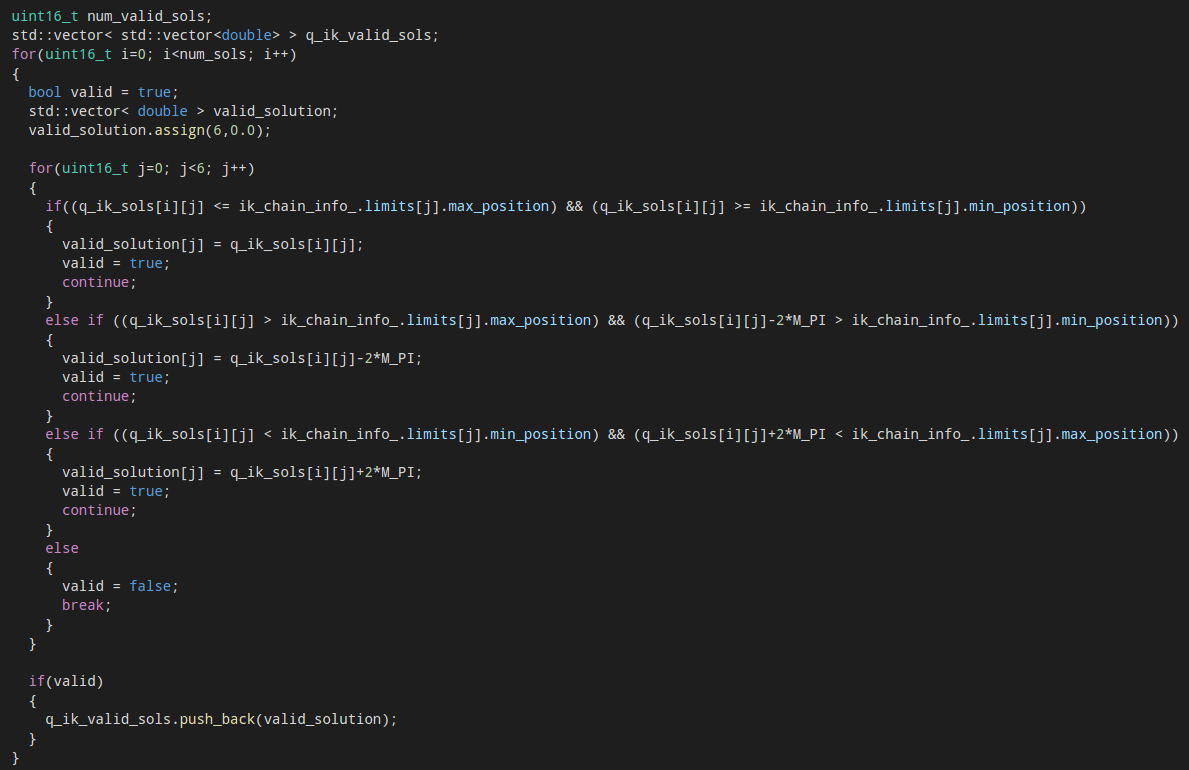
This method is contained in searchPostionIK() function.



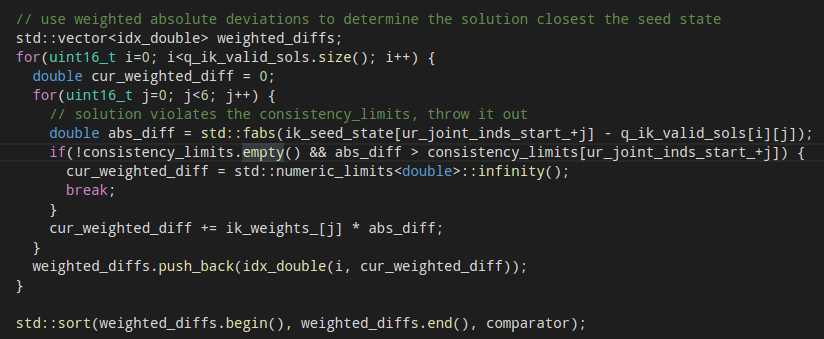
We create a homogeneous pose and put it in the 4\*4 array. Then we create a q\_ik\_sols[8][6] to contain the 8 solutions of inverse kinematics.

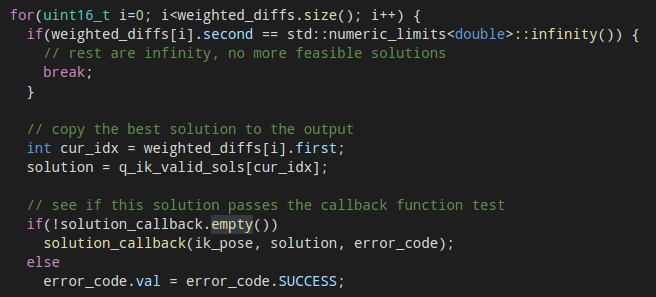


Do the analytic IK.



Verify if each solution is valid.





Choose the best solution closet to the current state.

**Step 5:** Chooseur\_moveit\_plugin in moveit\_setup\_assistant

