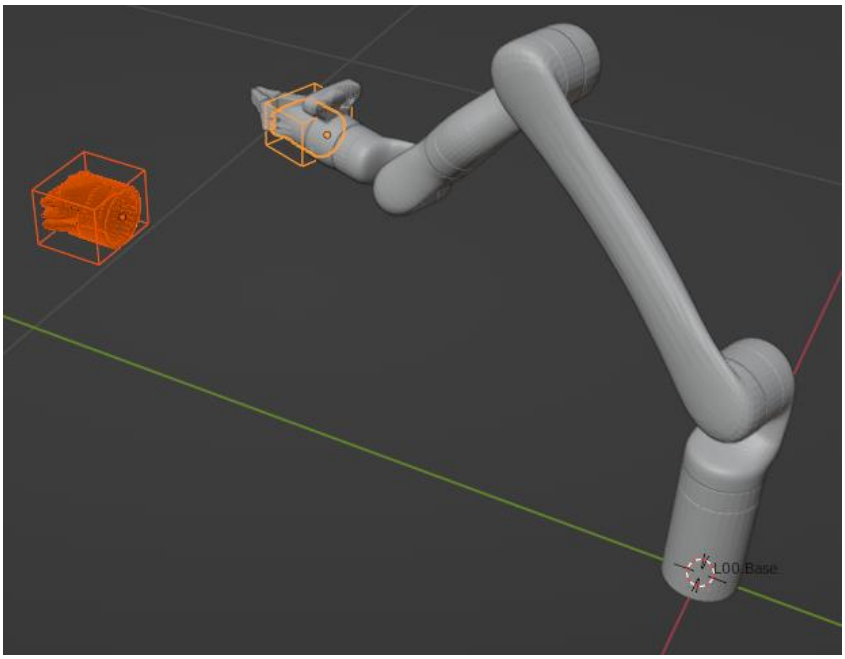
POLYTECH
NICE-SOPHIA

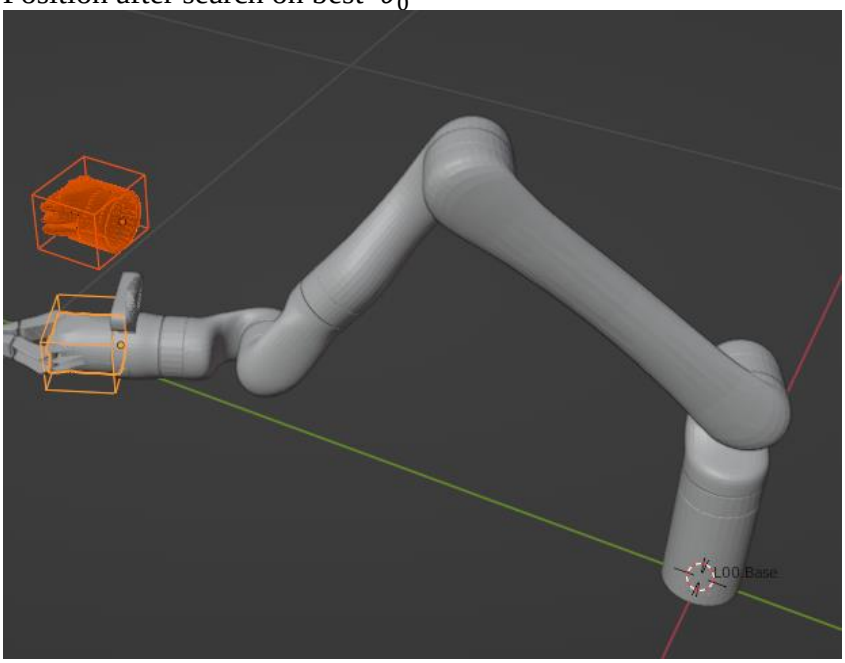
Project2: Help on Computing IK for a Robotic Arm

Progress step by step: After finding the best position for θ_0 (rotation of the base), you exit the loops. As such the program will generate the blender script and you will be able to validate you code in incremental steps.

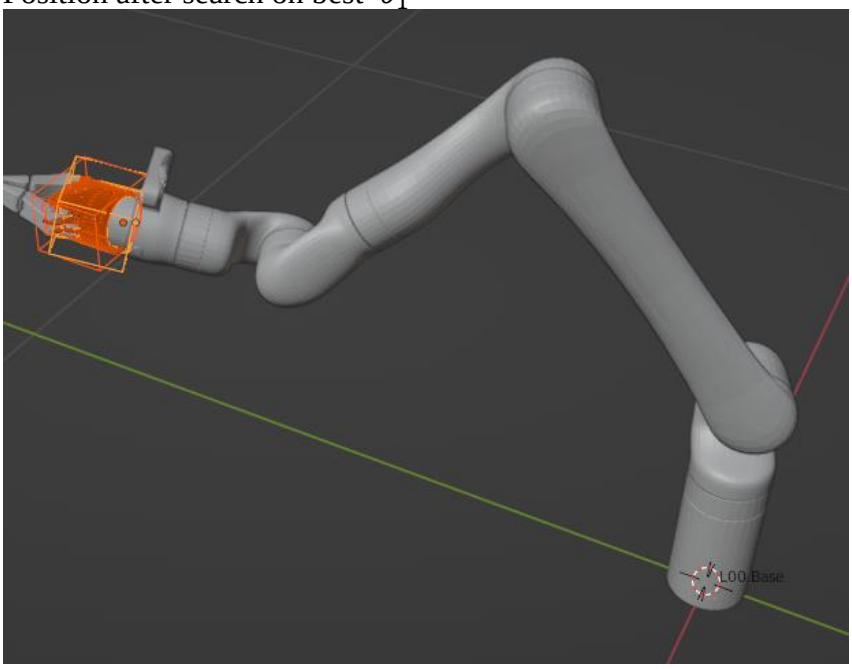
Initial Position



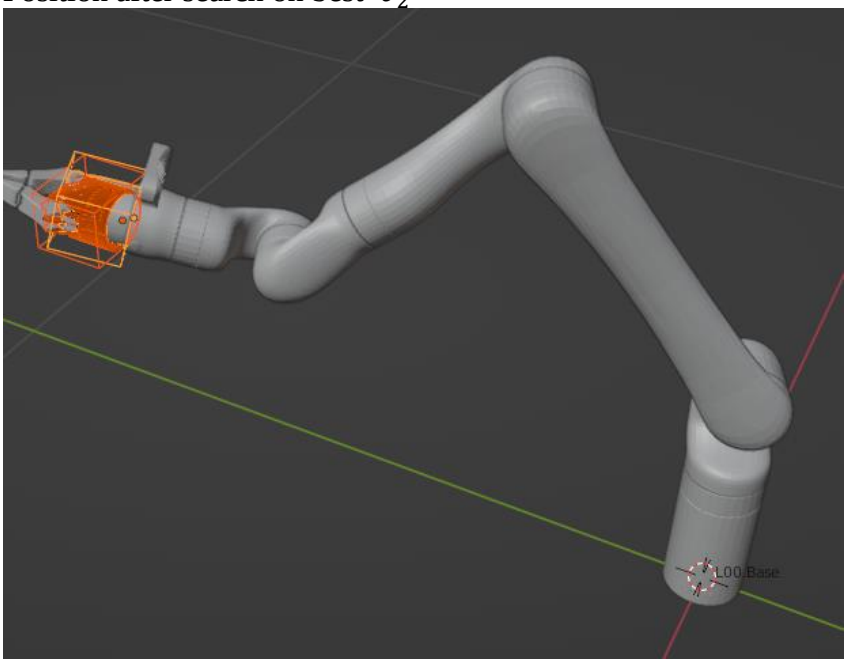
Position after search on best θ_0



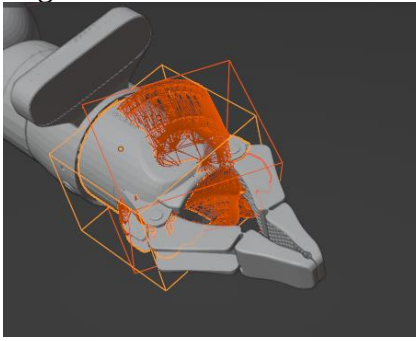
Position after search on best θ_1



Position after search on best θ_2

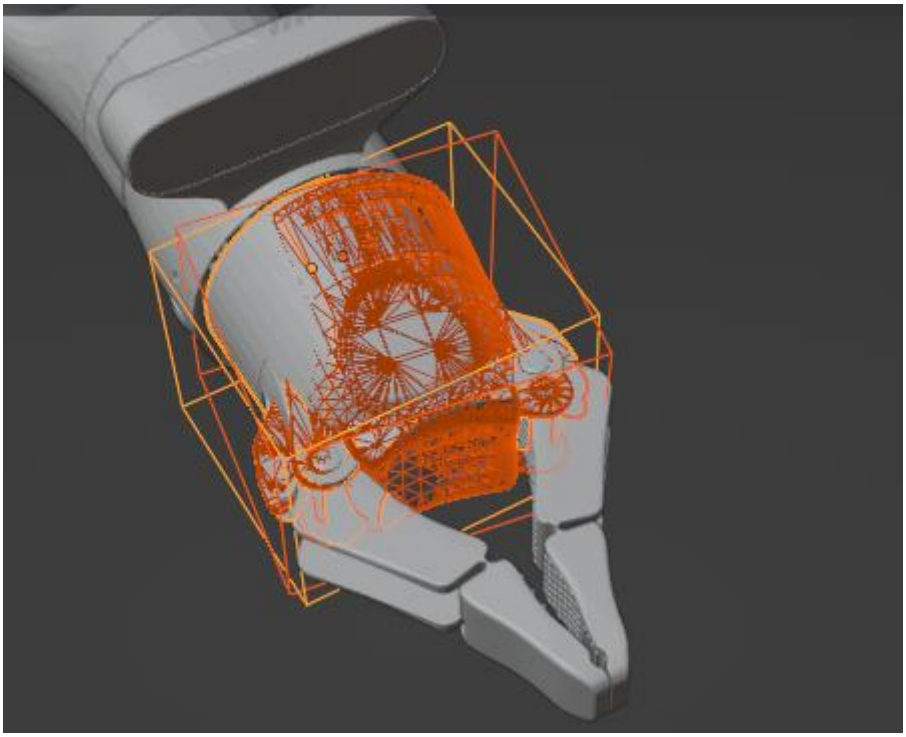


Limited changes are observed between these last two moves, The position is already close to the target reference. Close but not acceptable, see how the 2 b-boxes are mis-aligned.

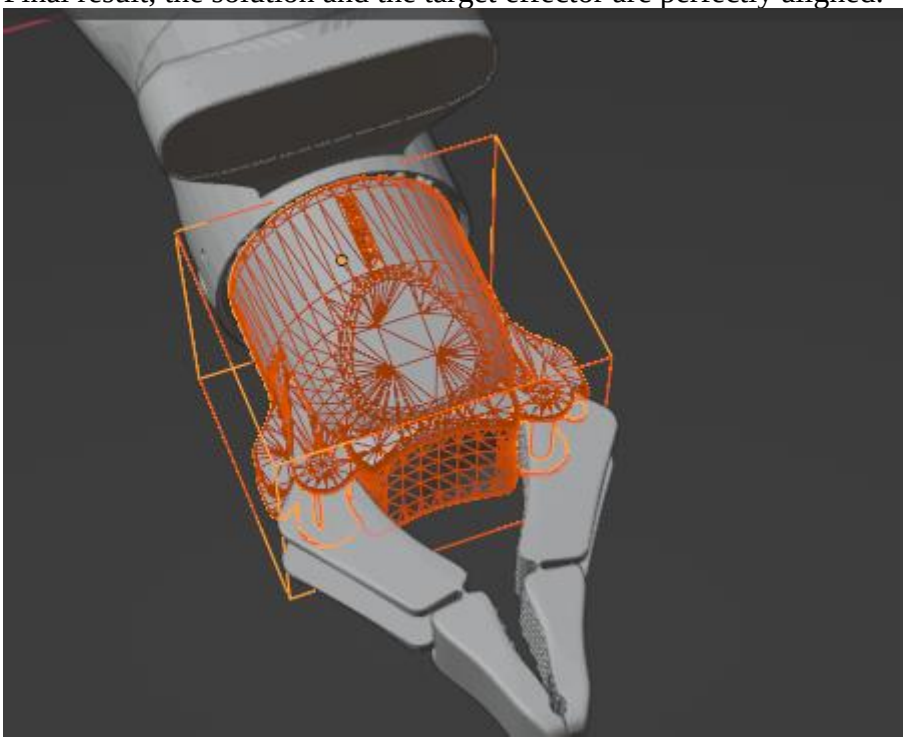


Depending on your strategy for the angle, convergence could be slow.

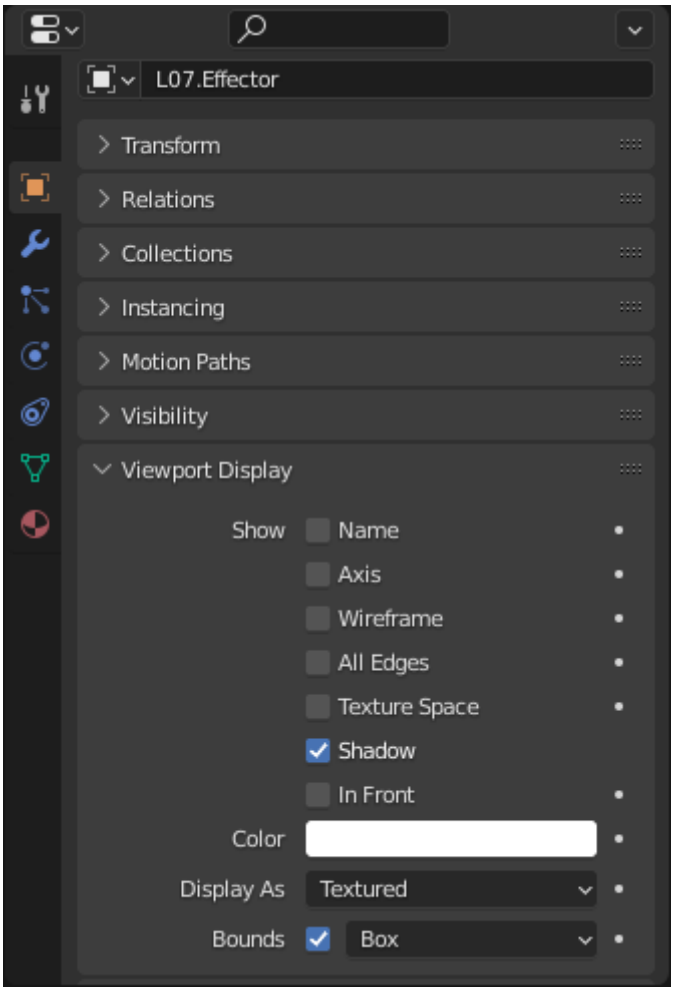
Intermediate result



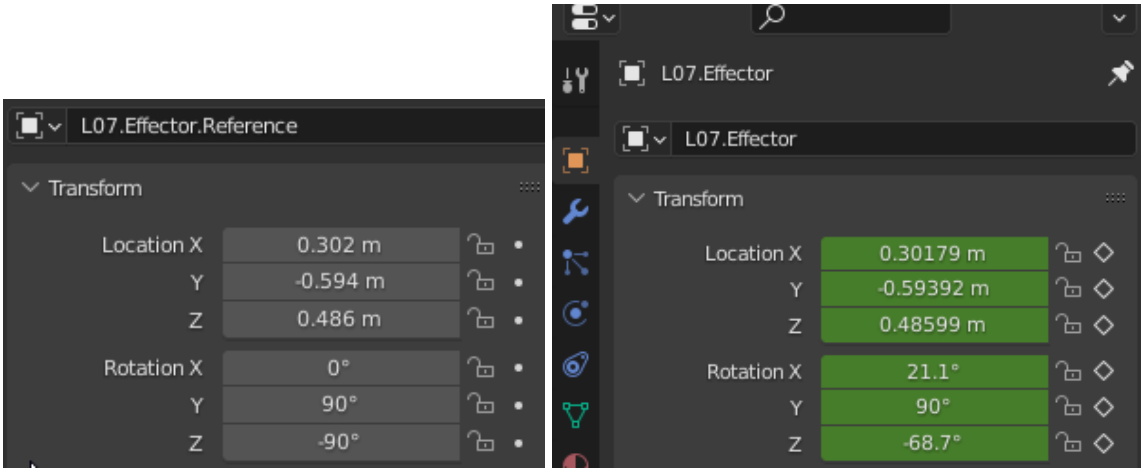
Final result, the solution and the target effector are perfectly aligned.



For better visibility, I have enabled the bounding box to be shown in blender



Good convergence is given by comparing the reference effector transform with the solution effector transform

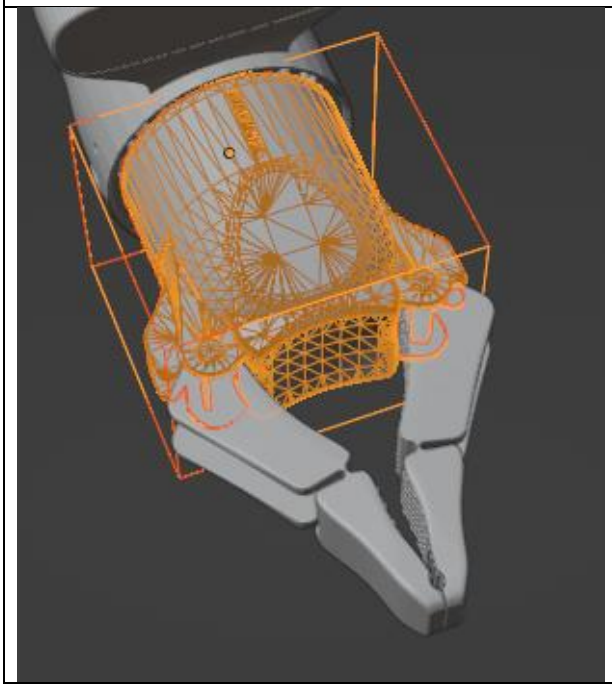
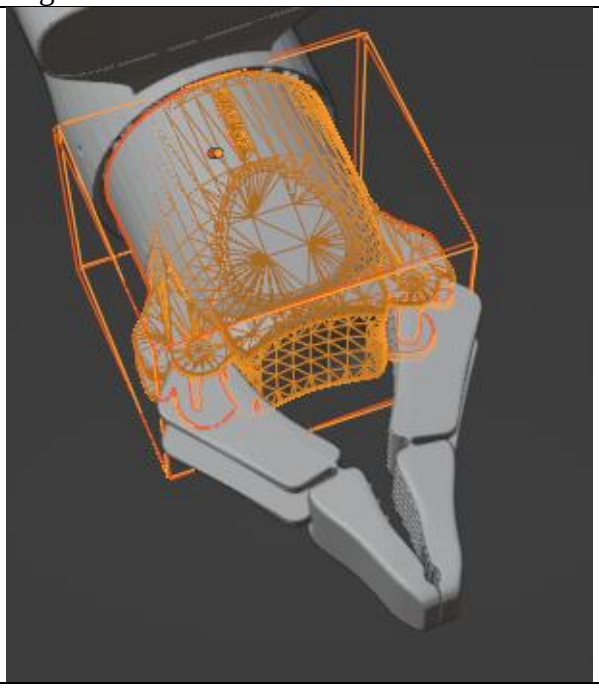


Although the rotation XYZ differs, you can note that Z-X = -68.7 – 21.1 = 89.8, very close to -90 for Z.
When Y is 90, multiple equivalent rotation matrices are possible.

The joint angles for this solution are:

```
(-125.332,-35.3387,79.6959,-58.8867,-42.1257,-39.1388)
```

Note the final result when the angle resolution changes (see next page for more details)

16 bit resolution: no visible mis-alignment	14 bit resolution: small but noticeable mis-alignment
	

Angle resolution is a major characteristic of the robotic arm for precision control.
High precision angular position sensors encode the angle with a 16 bit number, that about 0.0055 degree of resolution. Good quality angular sensors encode the angle with a 14 bit number, that 0.022 degree of resolution.
No need to go below the 16 bit resolution value, 14 bit resolution is fine.

Once you have reached good convergence, you can start focusing on the speed:

Key idea: as the number of iterations increases, you will have to search for less points but with greater resolution.

Deliverables for this project

An archive with your code (the content of the `source` directory).
A pdf file with screen capture from blender showing your results as well as some explanation on your IK solution, how you have computed the distance, which variant of CCD you have implemented. Include the command line that you have used to run your program.

How you will be graded:

- I'll compile your code. and run it with different targets, making sure for both FK and IK you obtain results similar to mine.
- I'll cross check your report to see your screen captures and your explanation.

Make sure that your screen capture and the code you deliver are aligned.