

Operational Control with iCub

Prerequisites

By now, you should be an expert of the following components :wink:: - Joint Control. - Cartesian Control. - Gaze Control.

Assignment

We want you to develop a module that employs the `Joint Interface`, the `Cartesian Interface` and the `Gaze Interface` to accomplish the following tasks:

1. Make iCub **look down at the table**.
2. **Detect the object** lying on the table.
3. Retrieve the **object position** in the Cartesian domain.
4. Let iCub **look at the object**.
5. Based on the object position, **select the best hand**.
6. **Control the iCub fingers** to achieve suitable grasp configurations.
7. Let iCub **approach the object** to enable a top grasp.
8. Ask iCub to **grasp the object**.
9. **Lift the object**.

The outcome should look like the animation below with the simulator:

Some of the points reported above have been already addressed in the code (e.g. **object detection and location retrieval**), so you need to **fill in the missing gaps** highlighted by the comment `// FILL IN THE CODE` in the `src/main.cpp` module.

Once done, you can test your code in two ways along with the simulator:

1. **Manually**: running the *yarpmanager scripts* provided from within `app/scripts` and yielding corresponding commands to the *module rpc port*.
2. **Automatically**: running the script `test.sh` in the `smoke-test` directory. Take into account these important points:
 1. We use a **timeout of 240 seconds** to check the status of rpc communication, meaning that you have *240 seconds* max to accomplish each rpc command.
 2. When you reply to rpc commands, we assume the robot has **finished the movement**.
 3. The smoke-test will add a random displacement to the initial position of the ball in order to force the use of both hands.