1 Tutorial

Here we give a quick tutorial to allow you to use the program.

1.1 Pre-Installation

You need do have installed YARP, ICUB, GAZEBO, WholeBodyDynamicsTree.
You also need to install the geomagic touch, see here: https://github.com/inria-larsen/icub-manual/wiki/Installation-with-the-Geomagic-Touch

1.2 Tutorial to launch the Haptic Device program

1. Connect your driver and choose the Ethernet network that corresponds to your geomagic.
2. Add the environement of the geomagic before any configuration and utilisation by copying:

```
export GTDD_HOME=/opt/geomagic_touch_device_driver
export LD_LIBRARY_PATH=/opt/geomagic_touch_device_driver/lib
export QT_PLUGIN_PATH=/opt/geomagic_touch_device_driver/lib/plugins
```

You can find these lines in */CppProgram/configFiles/geomagic.sh*. We have to create these environement variables each time, because otherwise this QT version enters in conflict with the iCub's QT version.
3. In a "yarpserver" terminal, launch yarpserver.
4. In a geomagic terminal, Run

```
/opt/geomagic_touch_device_driver/Geomagic_Touch_Setup
```

Normally, if your computer is well connected to the geomagic touch, the device model proposed in the interface is Geomagic Touch (if not, try to connect again your geomagic). Then, click on "pairing" and click on the geomagic button on its back.
Run:

```
/opt/geomagic_touch_device_driver/Geomagic_Touch_Diagnostic
```

Click on the right arrow until the calibration is done.
5. Launch:

```
yarprobotinterface --context geomagic --config geomagic.xml
```

A couple of ports are created, you can see the device state (a vector of size 8: position, orientation, button1, button2) by reading the port "state:o".

1.3 Tutorial to launch programs and ports linked to Gazebo

6. In a gazebo terminal, go back in the README folder (where there is the worldPROMPS.sdf file) , and launch gazebo using:

```
gazebo -slibgazebo_yarp_clock.so worldPROMPS.sdf
```

You can see some sphere that represents the three goal the robot learnt to reach in our example, and another one that represent the initial position of the arm robot.
7. in a dynamics terminal you have to launch the dynamic computation by writing:

```
wholeBodyDynamicsTree --autoconnect --robot icubGazeboSim
```

8. In a "kinematics" terminal, launch the kinematics computation software of the robot for the left arm (since we learn it's left arm movement):

```
iKinCartesianSolver --robot icubGazeboSim --part left_arm
```

9. In a "simCart" terminal, launch the simulation of the cartesian control of the robot using:

```
simCartesianControl --robot icubGazeboSim
```

1.4 Tutorial to record trajectories (C++ program)

10. Now if you go in the *CppProgram/build/bin* folder, you can launch:

```
./record
```

11. Now, you have to connect the port that gives forces information (from wholeBodyDynamicsTree previously launched in 7) to the Cpp port by launching in a new terminal:

```
yarp connect /wholeBodyDynamicsTree/left_arm/cartesianEndEffectorWrench:o /record/read
```

Then you can control the left arm movement of the iCub thanks to the geomagic.
Moreother, you can record the trajectories by pressing during the whole movement the first geomagic button (the darker one).
These trajectories will be recorded in the file record.txt.

1.5 Tutorial to learn and plot distribution other trajectories (Matlab program)

1. You have to prepare the learning: use the previous "record trajectory" program.
In our case, we start from the sphere near to the robot, and finish in one of the other points (see video).
For each kind of trajectory, create a folder (for example traj1) in MatlabProgram/Data and put all the recordX.txt files that you have created thanks to the "record.cpp" program.
In the MatlabProgram folder, open demo_plotProMPs.m file. In it, put into the parameter "nameDataTrajectories" the path of your folder (for example 'Data/traj1').

2. Now you can allow the robot to learn:
In a "matlab" terminal, launch Matlab and goes into the folder MatlabProgram.
Now launch demo_plotProMPs.m file, you will see the plot of the trajectories you have done, in another plot the distribution learned and in a last one an inference example of an initiated trajectory.

1.6 Tutorial to replay trajectories (matlab and C++ programs)

First, do the same actions than previous but this time for the demo_replayProMPs.m file (e.g. change the nameDataTrajectory parameters).
2. Launch the program demo_replayProMPs.m
If all the Yarp library are well configured, with the good path in Matlab, the following message will appear:

```
Yarp library loaded and initialized
Going to open port /matlab/write and read
Please connect to a bottle sink (e.g. yarp read) and press a button.
```

It indicates you to launch a Cpp program to replay the learned movement in Gazebo or to recognize a movement that you will initiate. Do:
1. Steps for in the Gazebo tutorial part.
2. Now if you go in the *CppProgram/build/bin* folder, you can launch:

```
./replay
```

3. In another terminal creates the different ports connections:

```
yarp connect /matlab/write /replay/read
yarp connect /replay/read /matlab/write
yarp connect /wholeBodyDynamicsTree/left_arm/cartesianEndEffectorWrench:o
/replay/readForces
```