1   Tutorial

Here we give a quick tutorial to allow you to use the program. A video tutorial is also available here:

`https://www.youtube.com/watch?v=OFIA7bvoaT8`

Note: we will give a name for all new terminals you will have to open. We advise you to rename all terminals with these names to allow you to find your way amongst all of them.

Note2: We call $YourPath$ the path where this software is installed.

Note3: We will not detail how works all these modules. Here, you will just be able to use the software by your own.

1.1   Pre-Installation

This software requires: YARP, ICUB, GAZEBO, WholeBodyDynamicsTree, yarpmanager. 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   Launching the Haptic Device

Once you have installed the Geomagic Touch device, follow these steps:

- Connect your driver and choose the Ethernet network that corresponds to your Geomagic. It seems that you need to close all other Ethernet or Wi-Fi connections to use it correctly.

- In a *yarpserver* terminal, launch Yarpserver.

- In a *server* terminal, run a server: $yarprun - -server/icub01$

- In a *geomagic* terminal, add the environment variables required by the Geomagic by copying the lines written in $YourPath/CppProgram/configFiles/geomagic.sh$:

  ```
  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
  export LC_NUMERIC=en_US.UTF-8
  ```

  We have to create these environment variables each time, because otherwise this QT version enters in conflict with the iCub's QT version.

- In this same *geomagic* terminal, launch:

  ```
  /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". Often, when it is not the case, it is because you are not well connected to the Geomagic.

  Click on *pairing* and push the Geomagic button on its back (the Ethernet cable). When it is paired successfully, you can close this window by clicking on *apply* and *ok*.

- Run:

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

  Click on the *next* button (right arrow) until you completed the calibration tab.

- Now, run:

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

  A couple of ports are created. Type the command *yarp name list* to read their name. By using the command *yarp read geomagic/state:o*, you can read the device state. This state corresponds to a 8D vector composed of: "position, orientation, button1, button2" values.

1.3 Launching modules and ports for the Gazebo simulation

- In a *gazebo* terminal, go to $/YourPath/CppProgram/configFiles/myWorld$, and launch gazebo using:

```
gazebo  -slibgazebo_yarp_clock.so  world2height.sdf
```

  The world *world2height.sdf* is an example where you have two balls that represent two goals the robot has learned to achieve from a Matlab script demo (*demo_plotProMP.m*).

  You can replay this world by the one you want. For example, we propose some worlds where other goals are represented within the *myWorld* folder.

Copy and past the ".ini" files that are in the folder $YourPath/CppProgram/configFiles$ to your *home* directory.

If you have trouble when you launch programs, it can be because of these ".ini" files. Indeed, the automatic "cartesianSolver.ini" and "simCartesianLeftArm.ini" files are written for the "iCubSim" simulator instead of "icubGazeboSim". This is why we give you other *.ini* files in *configFiles* with the good configurations for the iCubGazeboSim simulator. In that case, verify you place correctly these *.ini* file in the *home* directory, and re-run yarpmanager.

- From a *yarpmanager* terminal, launch yarpmanager.

  Then click on the button *Open file* and open $YourPath/App/recordTrajectoriesGeomagic.xml$.

- Go to the *Application* folder of *yarpmanager*, and double click on the *recordTrajectoriesGeomagic* application.

- Connect the different devices *WholeBodyDynamicsTree*, *iKinCartesianSolver* and *simCartesianControl* one by one and wait a little between each step.

1.4 Recording trajectories (C++ and Geomagic)

- In a *Cpp* terminal, you have to build the CppProgram project by:

```
cd $yourPath$/CppProgram
mkdir build
cd build
ccmake ..
make install
```

- In the *Yarpmanager* GUI previously opened, run the *recordTrajectoriesWithGeomagic* module, and click on *connect* to connect the required ports. By default, the recorded files will be saved in your home folder. To change this default folder, open the $YourPath/App/recordTrajectoriesGeomagic.xml$ file, and modify the line 36 as following:

```
<parameters>--folder $YourPath/MatlabProgram/Data/NewTraj</parameters>
```

  You can then control the left arm of the iCub thanks to the Geomagic. Note that we keep the orientation fixed to prevent incorrect movements when we use this program on the real iCub.
  To record the trajectories, maintain the first Geomagic button (the dark one) pressed during the whole movement.

  These trajectories will be recorded in files *recordX.txt*, where *X* is the x-th trajectory, in the folder you have chosen before.

- You can close this module from the *Yarpmanager* GUI.

1.5 Learning and plotting ProMPs (Matlab program)

This module allows you to plot ProMPs' results: observed trajectory, learned distribution, and inference of an initiated trajectory. For this module, you just need to use Matlab.

- Let us assume you have recorded some trajectories using *recordsTrajectoriesWithGeomagic* module of yarpmanager. For each trajectory type, create a folder (for example traj1) in $YourPath/MatlabProgram/Data$ and put all its *recordX.txt* files on it.

  In our case, we start from the sphere near the robot, and finish in one of the other points, as shown in the following video:

  https://www.youtube.com/edit?o=U&video_id=OFIA7bvoaT8.

- In a *matlab* terminal, launch Matlab and go into the folder $YourPath/MatlabProgram$.

- Open the *demo_plotProMP.m* script. Modify the parameter "nameDataTrajectories" value with the path of your folder. You can keep the default value $Data/traj1$.

- Run this script. Some plots will appear, that represent:

  * trajectories you have done;
  * the distribution learned thanks to these trajectories (mean and 97.5 percentile of the normal distribution);
  * an inference example from some early observations of one of your recorded trajectories. Note that this trajectory was not use to learn the distribution.

1.6 Replaying ProMPs (Matlab and C++ programs)

Thanks to this module, you can replay learned distribution in iCubGazeoSim, and to infer the end of an initiated trajectory. Note that for this module, you don't need to use the Geomagic Touch device.

- Do the same actions than previous subsection, but this time for the *demo_replayProMPs.m* script.

  If all the Yarp libraries are well configured, 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.
  ```

Now you can launch a Cpp program to replay the learned movement in Gazebo or to recognize a movement from early observations.

- Do steps of the Gazebo tutorial part 1.3 without the connection to the Geomagic device (not required).

- From *Yarpmanager* Guy, launch the module *replayTrajectory*

- Connect the different ports by clicking on the *Connect* button.

- Follow the Matlab's terminal instruction, and keep an eye on the iCubGazeboSim: you will see the robot replaying the learned observation and to finish an initiated movement on its own.

1.7 Inferring the end of an initiated trajectory thanks to ProMPs (Geomagic, Matlab and C++)

Do the same than in 1.6, with these following specifications:

- Add the Geomagic module.

- Connect[1] the speech module to allow the robot to talk by addding the *yarpdev* module in *Yarpmanager*, or directly by writing in a terminal:
  $yarpdev--devicespeech--lingware-contextspeech--pitch70--speed70$

- Run the script *demo_replayPromPsWithGeom.m* instead of *demo_replayProMPs.m*.

- Run the module *replayTrajectoryWithGeom* instead of *replayTrajectory*.

The difference with the previous module is that it is you who guides the robot during the early observation before the inference. To do that, just follow the Matlab terminal instructions: when I will wait for you movement, maintain the dark button of the Geomagic during the beginning of the movement, and loose it when you want it finish the movement.

---

[1]If you don't want to use this module, you can also comment the line 98 *sayType*(...) in the Matlab program *demo_replayPromPsWithGeom.m*.