**QuatProg.py**

* Base program to multiply quaternions for the collaborative robot.
* Used to get basic multiplications with relevant angles.

**adjust\_ur\_pose.py**

* Base program to generate targets.
* Is not cyclic.
* Generates action clients in each iteration.

**cam\_vec\_tran.py**

* Transform reference axes with regard to the camera to use them in the robot.

**collabROB\_brain.py**

* Obtains information from QR tag and robot position sensors.
* Generates an action client for UR robot.

**node\_red\_run.py**

* Runs Node red.
* Kills Node Red when it closes.

**quatPlot.py**

* Base program to multiply. quaternions for the collaborative robot.
* Used to get basic multiplications with relevant angles.
* It draws the quaternion transformation to use them.

**quat\_cam.py**

* final code to transform vectors to the correct transformation of the reference axes from the quaternions to use them by the collaborative robot code

**rob\_adjust.py**

* Final code that adjusts robot position in a cyclic way, obtaining the right quaternion an asking the user input confirmation

**telegram\_broker.py**

* Creates the telegram connection by using Node Red
* Obtains information from telegram and ubidots
* Writes data to the database generated for turtlebot positions

**turtlebot\_brain.py**

* Obtains information from robot position sensors
* Generates an action client for turtlebot
* Obtains information from telegram and ubidots
* Sends information to telegram and ubidots
* Obtain information from the database generated for turtlebot position

**ur\_pose.py**

* Obtains information from robot position sensors from the /TF