Care-O-bot Manual

Extension for

Universal Robot UR5 and connector

Authors: Florian Weisshardt
Nadia Hammoudeh Garcia
Bernhard Waterkamp
Thiago de Freitas

Fraunhofer IPA

Institute for Manufacturing Engineering and Automation
Stuttgart, Germany

Last modified on October 28, 2013

Chapter 1

Universal Robot on Care-O-bot

This chapter is an addition to the Care-O-bot manual which can be found at https://github.com/ipa320/setup/raw/master/manual/Care-O-bot_manual.pdf and explains handling the Universal Robot UR5 arm on Care-O-bot.

1.0.1 The UR arm

1.0.1.1 Start-up the UR controller

You can turn on the UR controller by pressing the button next to the key. After pressing the button the button should light up in green and the UR controller should boot up.

Next, you will need to release the emergency stop and initialize the UR arm by following the user interface on the touch panel.

NOTE: You can only release the emergency stop if the UR controller is booted up.

1.0.1.2 Operating the arm

For operating the arm a ROS node needs to be started. This is done by the bringup launch file

```
roslaunch cob_bringup robot.launch
```

or separately with

```
roslaunch cob_bringup ur_solo.launch ur_ip:=<<IP ADRESS OF YOUR UR
   CONTROLLER>>
```

After that you can directly operate the arm by using the command_gui or send a FollowJointTrajectoryAction to the arm.

1.0.2 The UR connector

To extend the workspace there's a ur connector which is an external 7th axis to the arm to be able to operate on the front and back side.

1.0.2.1 Start-up the UR connector

The ur_connector should be powered and ready to operate once there is no emergency stop active.

1.0.2.2 Operating the UR connector

For operating the ur_connector a ROS node needs to be started. This is also done by the bringup launch file, which if started before should no be relaunched.

```
roslaunch cob_bringup robot.launch
```

or separately with

```
roslaunch cob_bringup ur_connector_solo.launch ur_ip:=<<IP ADRESS OF
    YOUR UR CONTROLLER>>
```

After that you can directly operate the ur_connector by using the command_gui or send a FollowJointTrajectoryAction to the arm.

1.0.2.3 Handle failure situations

This section will cover some failure situation which might appear and how to resolve these.

Limit switches: When the ur_connector reaches one of the limit switches,the procedure for re-enabling its operation is to hold the brake lever down and manually move the ur_connector towards the opposite direction.

After that you should press the *Recover* button on the command_gui or call the ROS service by the command line, as:

```
rosservice call /ur_connector_controller/recover
```

No movement: If ever the ur_connector reaches a state where you can not move it anymore. First try to press the emergency stop button, release it, and further proceed with the aforementioned recover step.

If that still not work, stop all the other running programs, and call the *elmo_position* tool that can be found in the ipa_canopen package. To run the tool proceed with the following command:

```
./elmo_position /dev/pcanDEVICENUMBER CANID
```

In which, the arguments should initially be set to */dev/pcan0* and *CANID* to 11.