Robotic Manipulation Course

Exercise 4

February 18, 2020

1 Assignment

The goal is to implement a controller which will compute torques for the joints such that the cartesian force/torque at end effector is constant and given by:

$$\boldsymbol{f} = \begin{pmatrix} 0 & 0 & -1.0 \end{pmatrix}^\top, \quad \boldsymbol{M} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^\top. \tag{1}$$

The controller is implemented via ROS control interface and the computation of torques is performed in function *update()* of the controller class *Exercise4Controller*. Note, that gravity term is added to the torques automatically. Therefore, you need to compute external torques only. The controller will be tested on two different robots:

- ideal robot - for which the gravity term is computed exactly,
- simulated real robot - with gravity term error.

The gravity term error might correspond to the calibration error and is typical when working with real systems. To test you controller, compile the workspace and use one of the following commands for ideal and real robot respectively:

```
roslaunch exercise4 sim.launch
```

```
roslaunch exercise4 sim_calibration_error.launch
```

Then run the following script:

```
rosrun exercise4 run
```

which will move the arm to the position above the floor and then will switch the controllers. The code collects the position and force profile of the end effector and puts the data into the file */tmp/exercise4.csv* in format:

`time, position_x, position_y, position_z, force_x, force_y, force_z, moment_x, moment_y, moment_z`

Plot the collected data to see if your controller works correctly. Implement and compare two versions of the controller:

- feedforward controller given by Eq. 11.50 in Lynch & Park,
- feedforward plus feedback-based (PI-controller) given by Eq. 11.51 in Lynch & Park.

Note, that gravity term $\tilde{g}(\theta)$ is added by underlying simulation automatically. Tune the PI-controller for the real robot such that applied force corresponds to the desired force (i.e. the steady state error is zero).

Both robots (ideal and real) can fall down after short period of time of force only control. This is an expected outcome because of interaction forces. In the analysis, we focus on the short time window after contact during which the controller is stable.

2 Report

In addition to code, you are supposed to write a technical report (pdf) in which you will document the steps performed to fulfill the assignment. Your report should contains:

- your name, student number, date, exercise number and course name
- the mathematical equation of torques computation
- plots of position/force profiles of the controllers for ideal/real robot
- discussion of the results
- answers to the following questions:
  - In which coordinate frame is the force/torque specified?
  - Is your force controller stable such that the robot remains in single stable robot configuration for a long time? If not, what technique can be used to stabilise it?
  - What is the benefit of the feedforward plus feedback-based controller compared to the feedforward only?
  - In which cases the integral term of feedback based controller is necessary?
- Estimate of time spent on this exercise

3 Submission

To submit your code and report, fork a repository named *robotic_manipulation_2020/exercise4* to your gitlab subgroup. Modify the code in the forked repository. Be sure to push your code before the assignment deadline. The commits pushed after the deadline will be ignored.

4 Deadline

Deadline for this assignment is 3rd of March at 23:59.

5 Resources

- ROS Control `http://wiki.ros.org/ros_control`