

Übungen zum Praktikum Control of Robot Manipulators

Final Project

Project

This exercise describes the end project, which will take from this week until the end of the semester. The goal of the project is to write a Player program that uses the right manipulator of the B21 mobile robot to build a tower with Lego bricks. The Lego bricks are located on a table in front of the B21 and the poses of all objects in the scenario are known. You have to first develop your program in the Gazebo simulator, where you have the advantage that the bricks will snap into each other, even if you do not connect them completely precise. The positioning error can be up to 1.6cm. If your program is robust and exact enough, you move on to the real B21. The bricks in the real world will be distributed like in the simulation or we will add a vision system that gives you the poses of the Lego bricks relative to a camera.

Setting

On the webpage you will find the file *brick_project.tar.gz*. It contains the following files

- worlds/empty_world/bricks.template
- worlds/empty_world/bricks.config
- worlds/empty_world/makefile
- worlds/empty_world/gazebo-includes/entities-processes.xml
- worlds/empty_world/player-includes/gazebo.config

Uncompress the archive and add the files to the corresponding directories of your kibo directory. You will have to *make bricks.template* to create *bricks.world*.

Project tasks

1. Bring the right manipulator to the following initial CSpace pose $\langle 2.0, -1.57, -0.3, -0.3, -0.3, 1.57 \rangle$ (the angles of the six joints are specified in radians). You will have to define intermediate manipulator states to get there, because if you send this command directly with `pi-actarray -i 119 -m -p -0 2.0 -1 -1.57 -2 -0.3 -3 -0.3 -4 -0.3 -5 1.57` the manipulator will collide with the table.
2. Use all bricks on the table and stack them at pose $\langle 0.3, -0.2, 0.0, 0.0, 0.0, 0.0 \rangle$ relative to the table. To do this, keep the following things in mind:
 - You can get the relative poses of all Lego bricks from the file *bricks.template*
 - Use via points for all movements of the manipulator
 - The via points should be generated automatically, e.g. every 2 cm or even more often
 - Use your inverse kinematics program from exercise 7 to compute the CSpace solutions for all via points

- Make sure that the manipulator moves as little as possible between two via points, so it should not change the joint angles too much. Especially the joints at the elbow are critical because they often lead to collisions
- Evaluate the quality of a trajectory, e.g. by computing the WSpace distance, that the endeffector has travelled, or the sum of all angle-changes over the whole trajectory