

Quick start

1. Simulation

1.1 Einstellung

- Die Position von dem Gegenstand in object_position.cpp (nach Änderung kompilieren)
- Die Startposition des Endeffektors in trajectory_planning.cpp (nach Änderung kompilieren)
- Geschwindigkeit in moveit_execution.cpp (move_group.setMaxVelocityScalingFactor()) (nach Änderung kompilieren)

1.2 Ausführung

- joint limits in
/home/jan-philipp/catkin_ws/src/fmauch_universal_robot/ur10_e_moveit_config/config
kommentieren.
- roslaunch ur10_e_moveit_config demo.launch limited:=true
- roslaunch trajectory_planning trajectory_planning.launch

2. Echter Roboter

1.1 Einstellung

- Die Position von dem Gegenstand in object_position.cpp (nach Änderung kompilieren)
- Die Startposition des Endeffektors in trajectory_planning.cpp (nach Änderung kompilieren)
- Geschwindigkeit auf dem Teach Pendant
- IP-adresse von Remoter und Roboter prüfen

1.2 Ausführung

1. joint limits in
/home/jan-philipp/catkin_ws/src/fmauch_universal_robot/ur10_e_moveit_config/config
unkommentieren.
2. roslaunch ur_robot_driver ur10e_bringup.launch limited:=true robot_ip:=192.168.56.2
kinematics_config:=/home/jan-philipp/catkin_ws/src/my_robot_calibration.yaml
3. Programm auf dem Teach Pendant (Program/URCaps) starten
4. roslaunch ur10_e_moveit_config ur10_e_moveit_planning_execution.launch limited:=true
5. Falls mit Rviz: roslaunch ur10_e_moveit_config moveit_rviz.launch config:=true
6. Alternative zu 4 und 5: roslaunch trajectory_planning planning_execution_rviz.launch
limited:=true
7. roslaunch trajectory_planning trajectory_planning.launch