

Programming Exercises 3.1 and 3.2

Kasper Høj Lorenzen

University of Southern Denmark

kalor@mmmi.sdu.dk

September 14, 2022

Overview

Programming Exercise 1.2

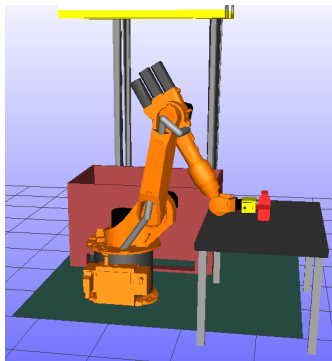
Programming Exercise 2

RobWork math and workcells

Programming Exercises 3.2

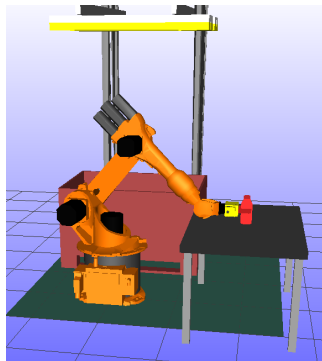
Programming Exercises 3.1

Programming Exercise 1.2



► $q = \{1.7712, -0.0548906, -2.41234, 2.88718, 0.912336, -1.413\}$

► $q = \{1.76941, -0.0565661, -2.41395, 2.88328, 0.910975, 4.71239\}$



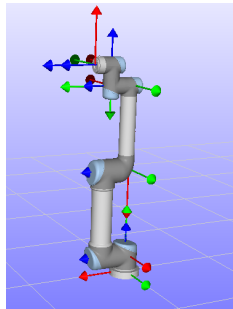
► $q = \{-1.42764, 0.66059, 1.56711, 6.05139, 0.670049, -1.38785\}$

► $q = \{-1.42899, 0.662597, 1.56841, 6.05942, 0.672475, 4.88856\}$

► Additional solutions if joint limits are relaxed

Programming Exercise 2

- Solution is on itslearning



RobWork Math

- ▶ RobWork includes types for all the transformations used in this course
- ▶ Take a look at the HelloRobwork program to see usage of the various transformations
- ▶ There is also a `Rotation3D` class
- ▶ Take a look at http://www.robwork.dk/apidoc/cpp/doxygen/namespacerw_1_1math.html and the Python API to see what other types there are

Loading a workcell with RobWork

- ▶ Workcells can be loaded into C++ and Python using RobWork
- ▶ See the pages for WorkCell and Device on www.robwork.dk

```
const string workcell_path = "/path/to/workcell/Scene.wc.xml";  
const string device_name = "device_name";  
WorkCell::Ptr wc = WorkCellLoader::Factory::load(workcell_path);  
Device::Ptr device = wc->findDevice(device_name);
```

Programming Exercise 3.2

- ▶ Program a function to calculate the forward kinematics
- ▶ Transform3D can be used to represent the transformations T
- ▶ Q can be used for the state vector q
- ▶ Compare your solution to the workcell KukaKr16 WorkCell

Programming Exercise 3.1

- ▶ Program a function to calculate the Jacobian
- ▶ The built-in type `Jacobian` can be used to represent the Jacobian $J(q)$
- ▶ Give the function a vector of Transforms, solve for the A and B part and merge it