Introduction to RobWork and Programming Exercises 1.1, 1.2 and 2

Kasper Høj Lorenzen

Univeristy of Southern Denmark

Kalor@mmmi.sdu.dk

September 1, 2022

Overview

TA

- Kasper Høj Lorenzen
- kalor@mmmi.sdu.dk
- Office Ø26-601b-3
- I'm usually in my office between 9:00am and 16:30pm

Format of the labs

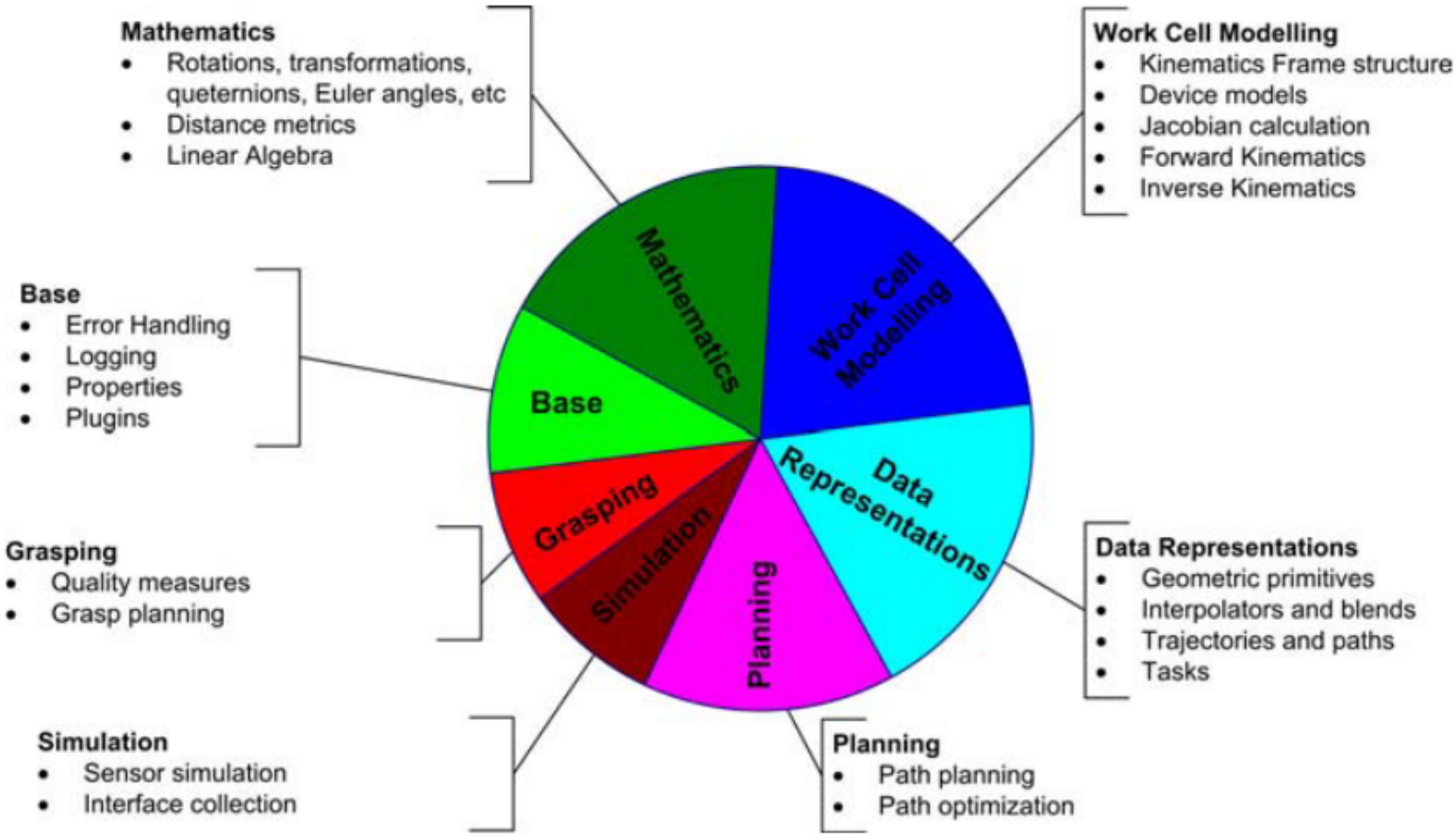
Structure of the exercises

- Explain the solution to last week's exercise
- Introduce this week's exercise
- Present hints or information needed to solve the exercise

What is RobWork

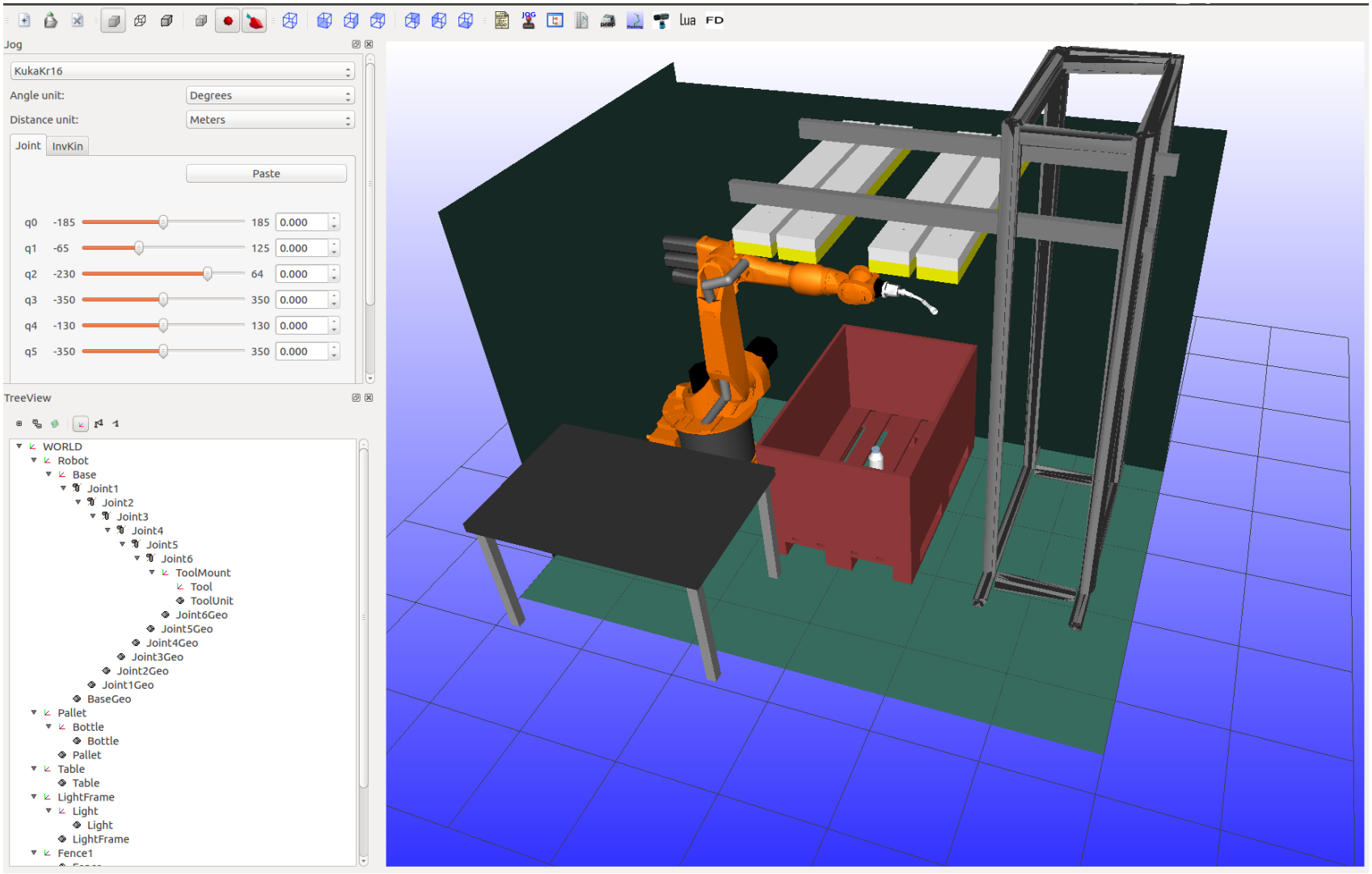
- RobWork is a collection of C++ libraries for robotics
- Developed at SDU
- Handles kinematics, path planning, collision checking, etc.
- Extendible via plugins
- Consists of four parts: RobWork, RobWorkStudio, RobWorkSim
- Documentation at `www.robwork.dk`

RobWork core content



[1] Image borrowed from [Ellekilde, Jorgensen, 2010]

RobWorkStudio

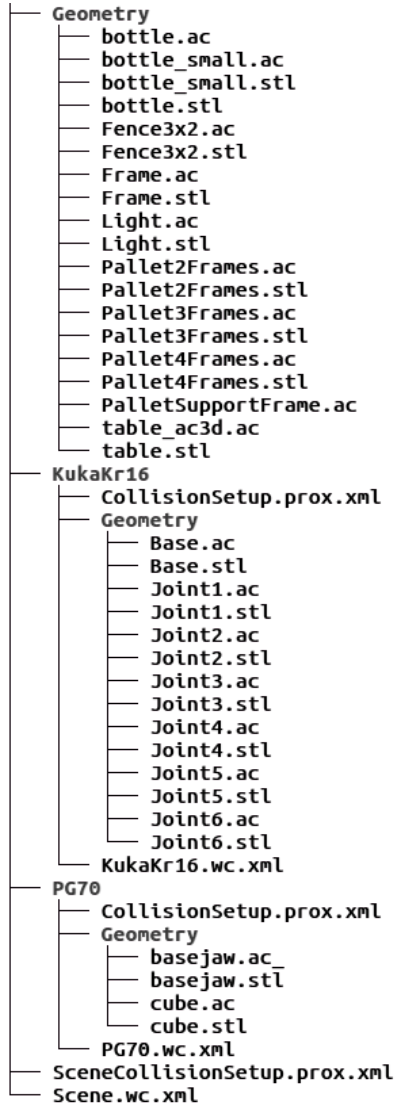


Tasks for today

- Compile and run the HelloRobWork program
- Programming Exercise 1.1 and 1.2
- Get familiar with RobWorkStudio and robot movement
- Use the jog plugin to move the robot
- Workcell is available on itslearning
- Visualize frames in the Tree plugin
- Do Programming Exercise 2
- Construct a RobWork workcell with a UR robot manipulator
- Geometries are from a CAD file
- Use datasheet (on itslearning) to get measurements
- Download workcell `UR5WorkCellCut.zip` from itslearning
- Edit the `Device.wc.xml` file

RobWork Workcell Structure

- A workcell consists of:
  - Geometries
  - Devices
  - Scene definitions (Frame definitions)
  - Collision Setup
- Each device is structured as a workcell
- More information can be found at `http://www.robwork.dk/file_formats/workcell/#`



RobWork XML files

- Frame definitions
  - Positions: x, y, z (red, green, blue) in $[m]$
  - Rotations: RPY $(\theta_z, \theta_y, \theta_x)$ in $[Deg]$
  - Type: Revolute or prismatic
- Joint limits: Have already been set
- Drawables
  - Graphics for a joint
  - `refframe` gives the coordinate frame for the graphics
  - Pose is relative to `refframe`
  - **WARNING:** The pose of the graphics objects is given in absolute coordinates w.r.t. the robot

Programming Exercise 2

- Guide to the first two joints.
- Based on slides by Lars Carøe Sørensen

Programming Exercise 2

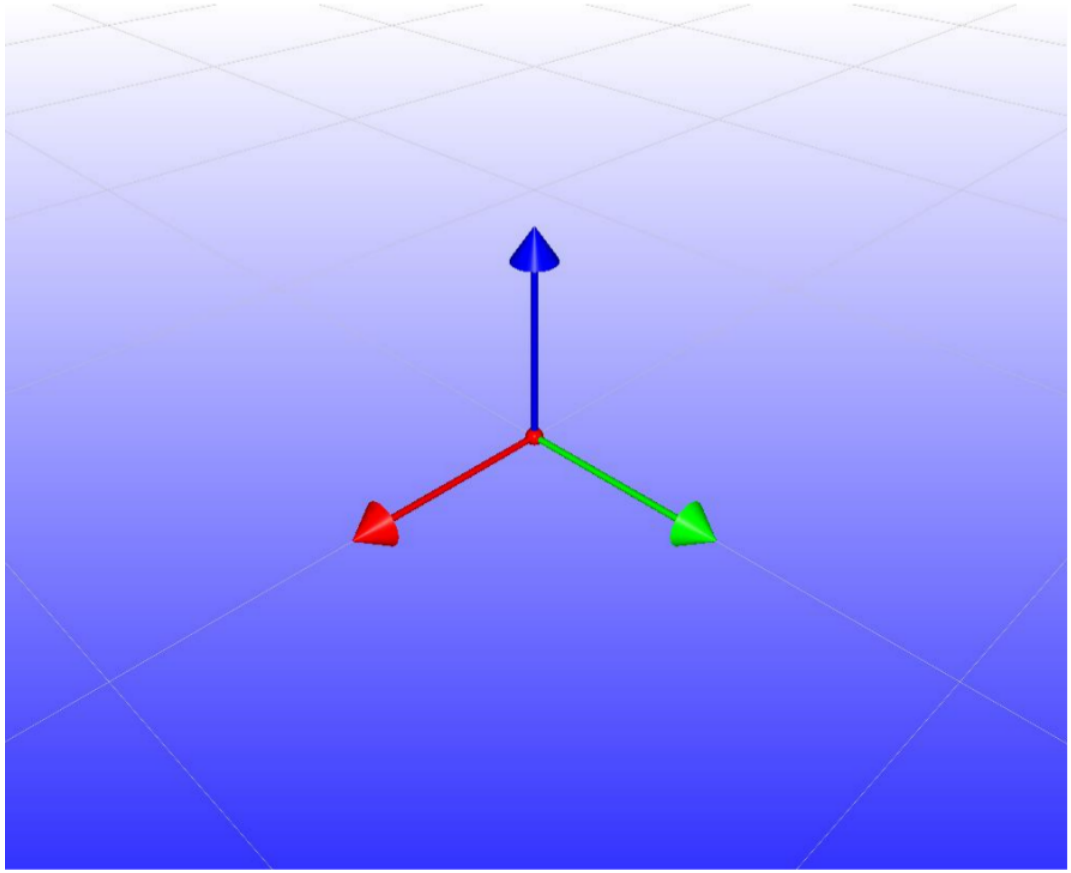
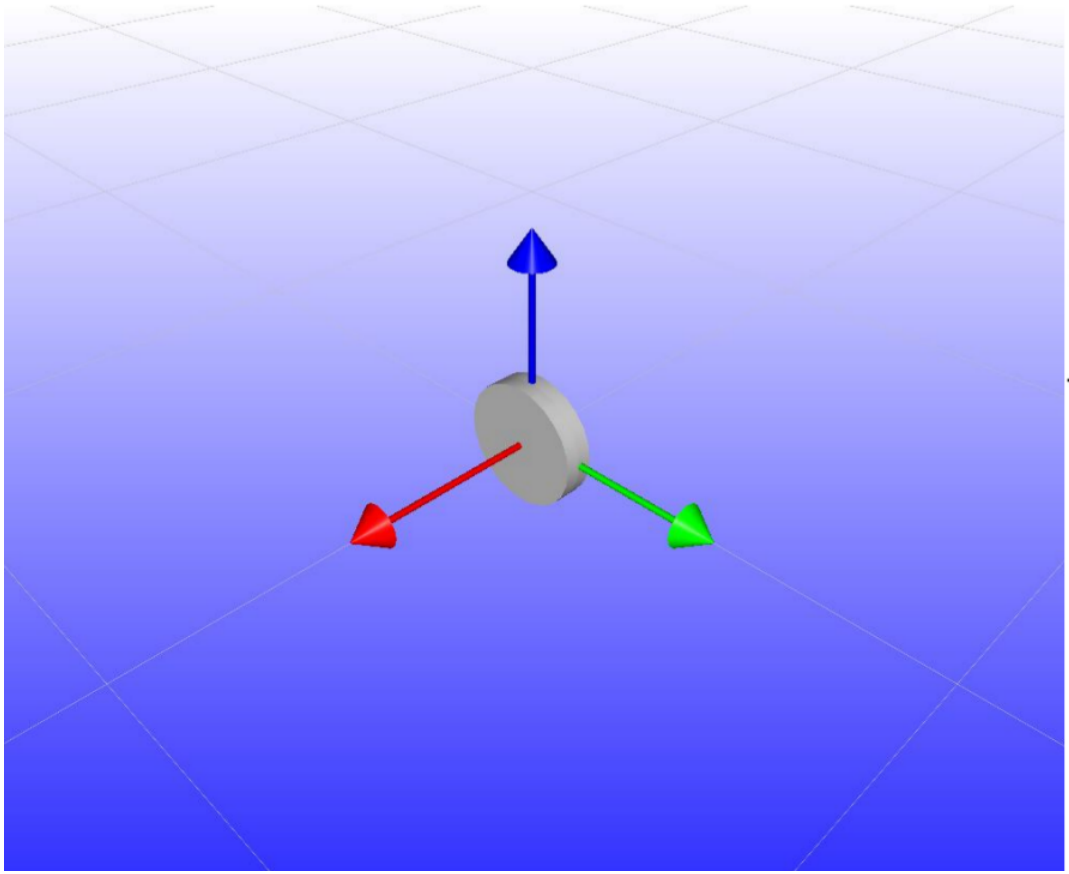


Figure: World/Robot/Base frame

Programming Exercise 2



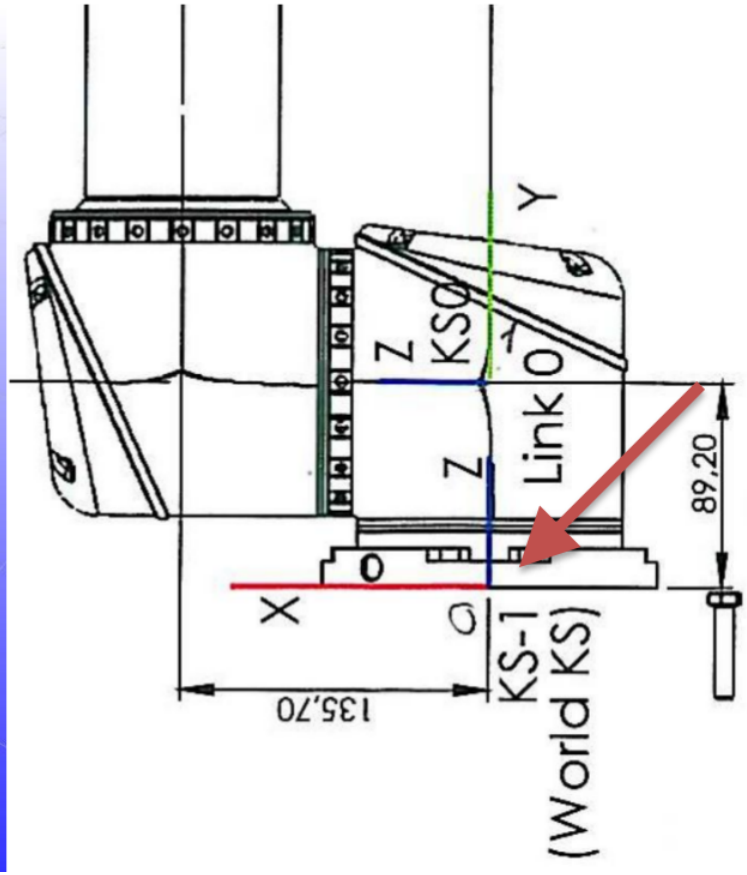


Figure: Insert robotFlange and base (all pos and rot zero)

Programming Exercise 2

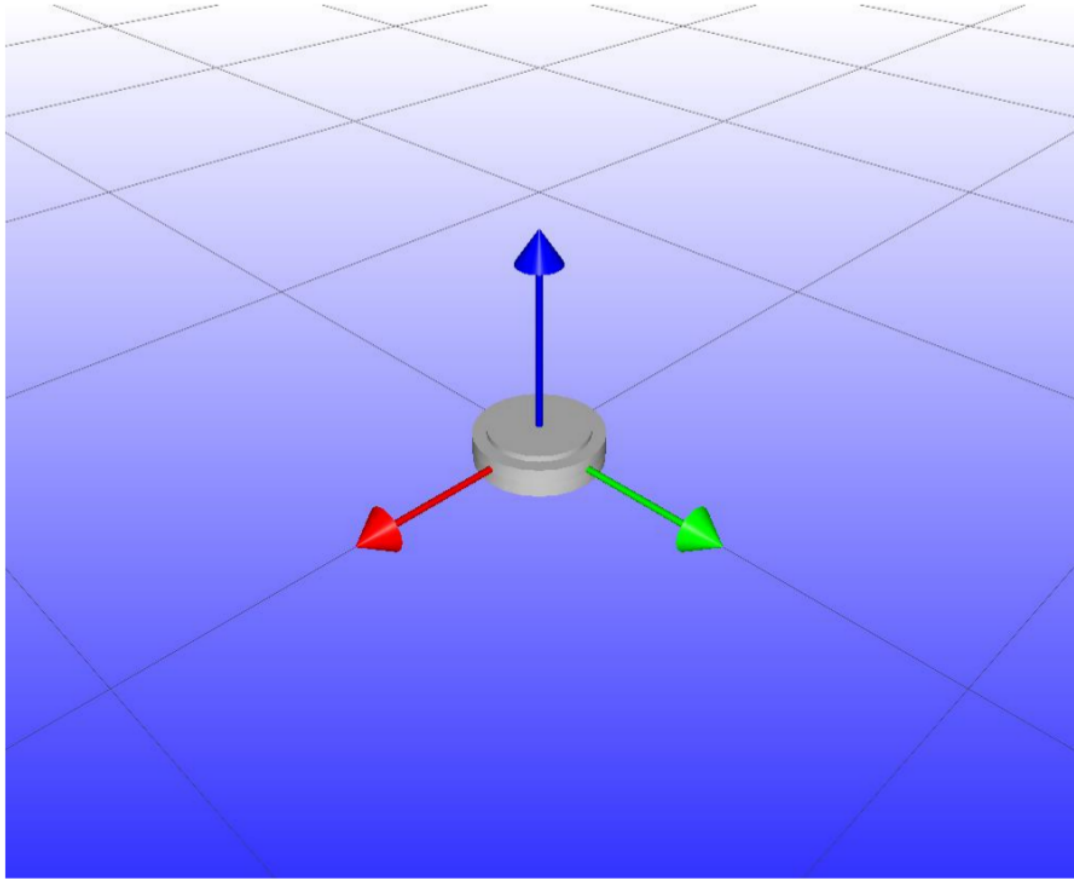
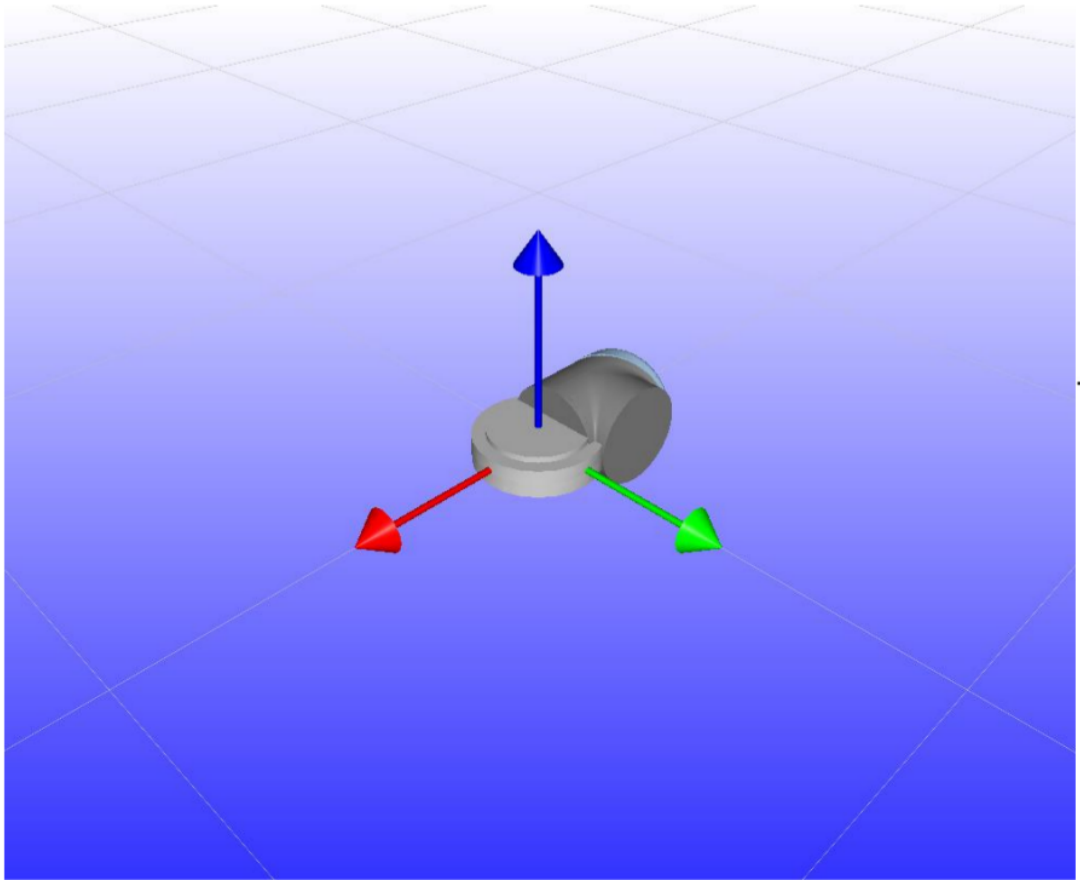


Figure: Drawable: rotate 90° about y ($P = 90^\circ$)

Programming Exercise 2

- Base and robotFlange in place. XML is:
- ```
  <Drawable name="flangeGeo" refframe="Base">
  <RPY> 0 90 0</RPY> <Pos> 0 0 0</Pos>
  <Polytope file="geometry/robotFlange" />
  </Drawable>
  <Drawable name="baseGeo" refframe="Base">
  <RPY> 0 90 0</RPY> <Pos> 0 0 0</Pos>
  <Polytope file="geometry/base" />
  </Drawable>
  ```
  ```

Programming Exercise 2



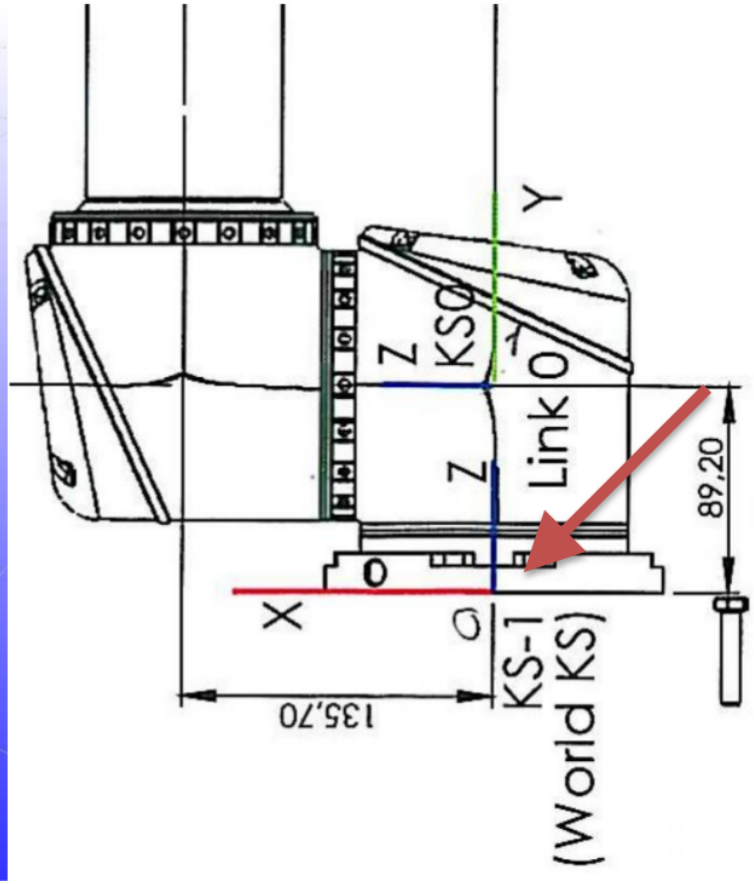


Figure: Insert Joint0 (all pos and rot zero)

Programming Exercise 2

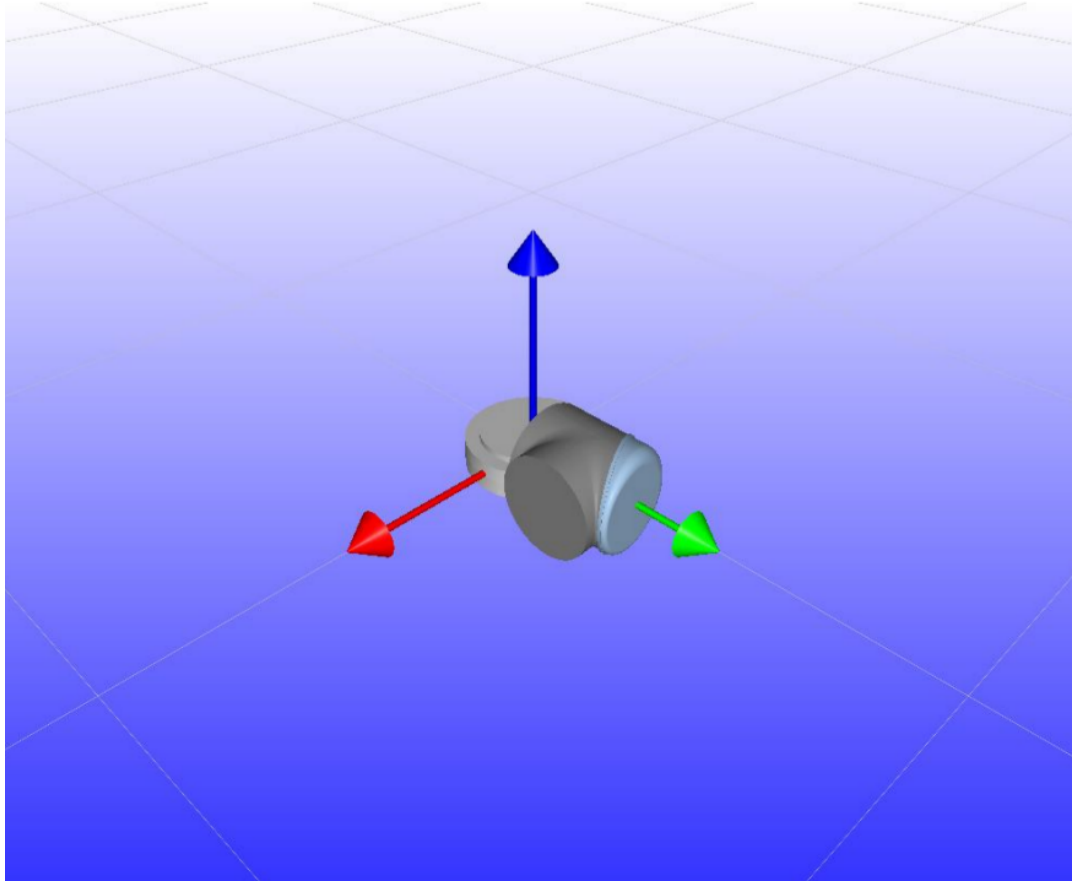


Figure: Drawable0: rotate 270° around z ($R = 270°$)

Programming Exercise 2

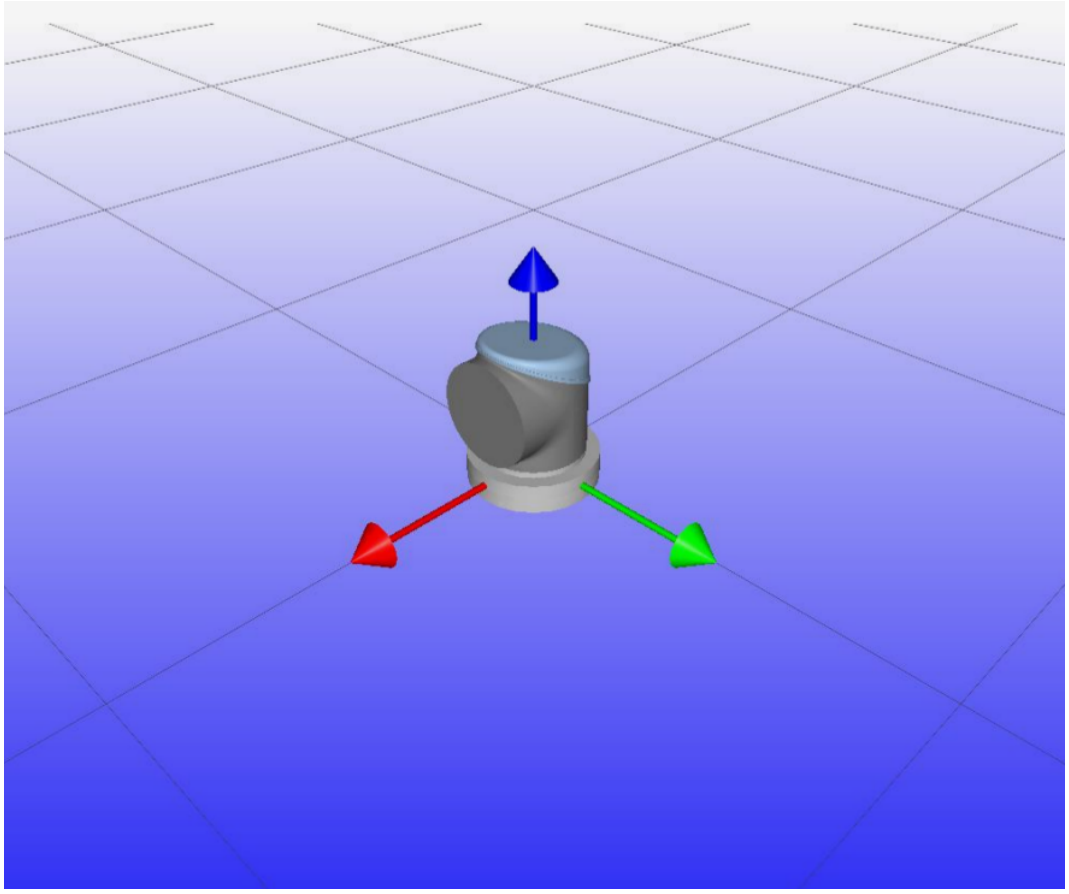


Figure: Drawable0: rotate 90° around y ($P = 90°$)

Programming Exercise 2

- Joint0 in place
- ```
  <Joint name="Joint0" type="Revolute">
  <RPY> 0 0 0 </RPY> <Pos> 0 0 0 </Pos>
  </Joint>
  <Drawable name="Joint0Geo" refframe="Joint0">
  <RPY> 270 90 0 </RPY> <Pos> 0 0 0 </Pos>
  <Polytope file="Geometry/joint0" />
  </Drawable>
  <Q name="Home">0</Q>
  ```
  ```

Programming Exercise 2

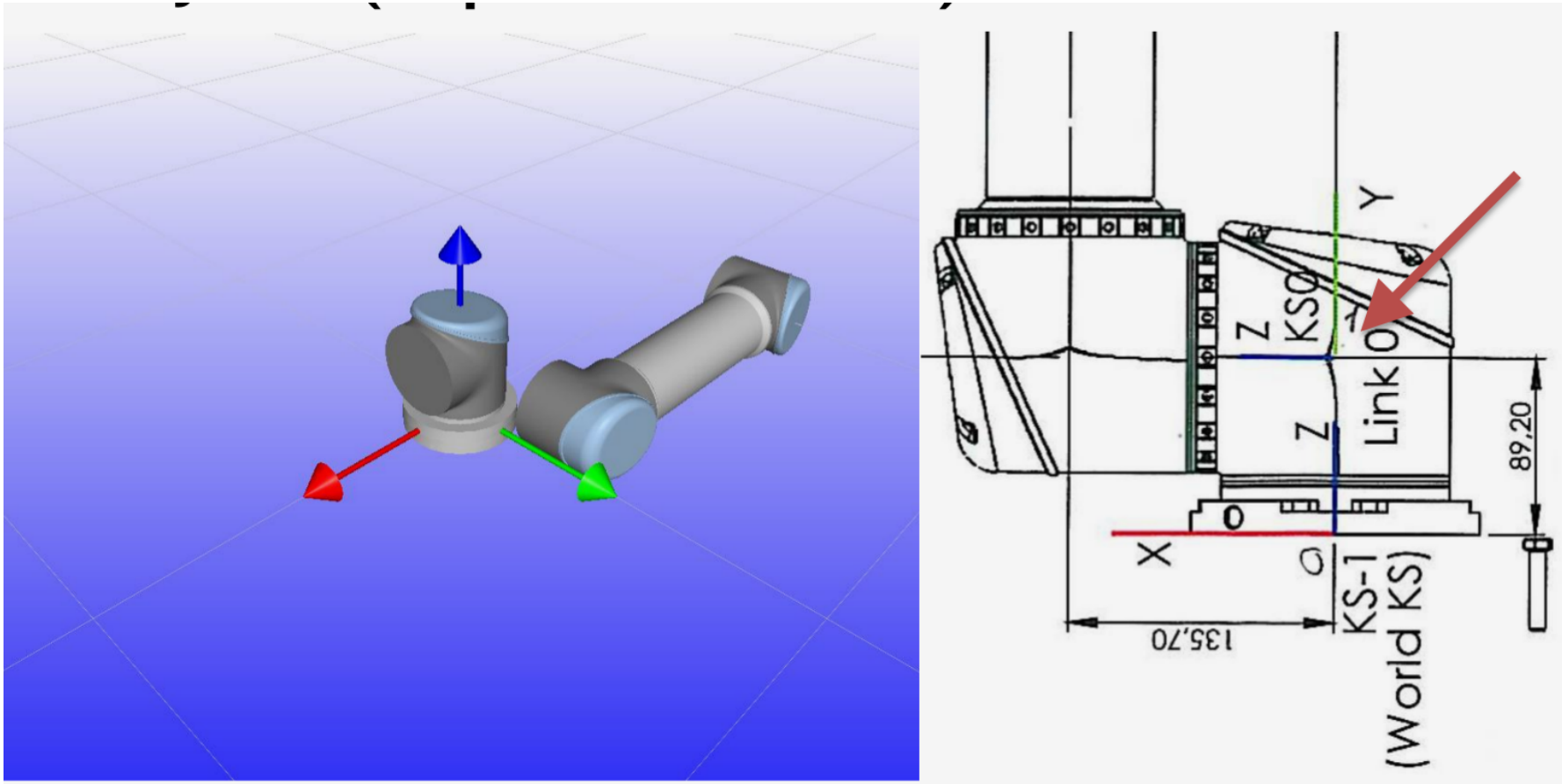


Figure: Insert Joint1 (all pos and rot zero!)

Programming Exercise 2

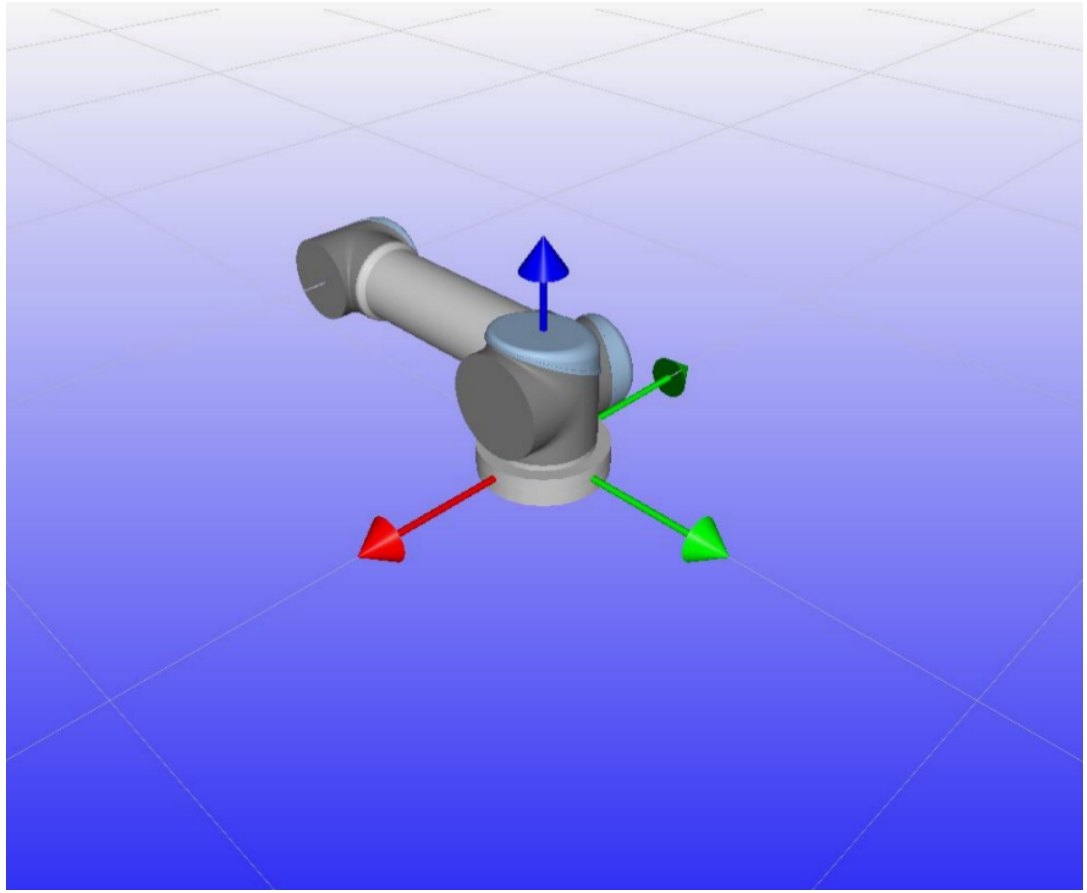


Figure: Joint1: rotate frame ($R = 90°$)

Programming Exercise 2

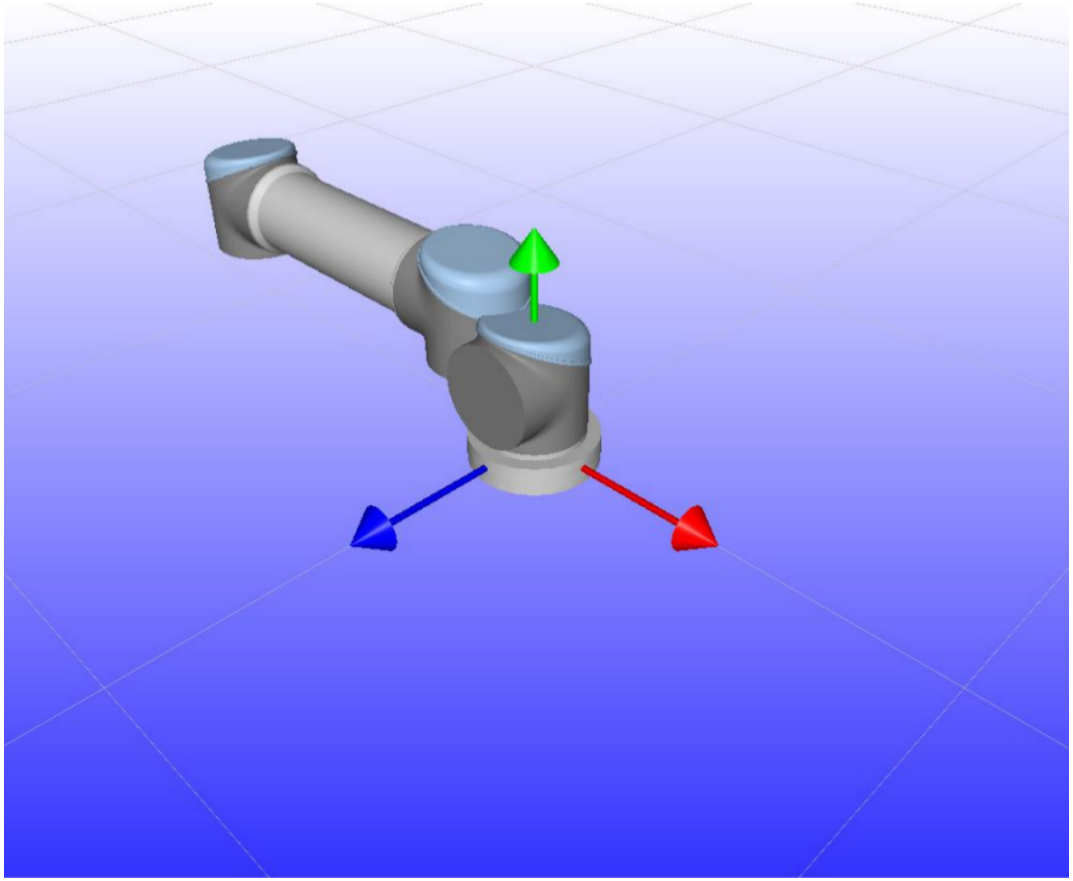


Figure: Joint1: rotate frame ($Y = 90°$)

Programming Exercise 2

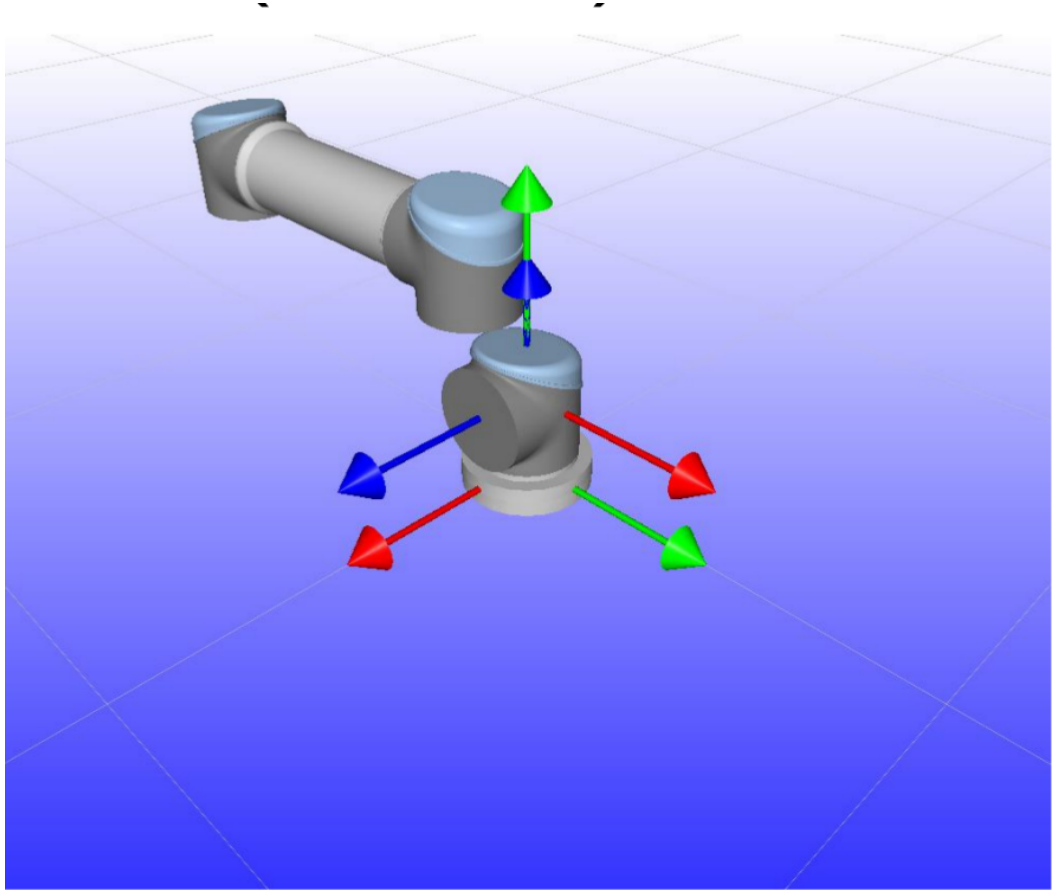


Figure: Joint1: move frame ($z = 0.08920$)

Programming Exercise 2

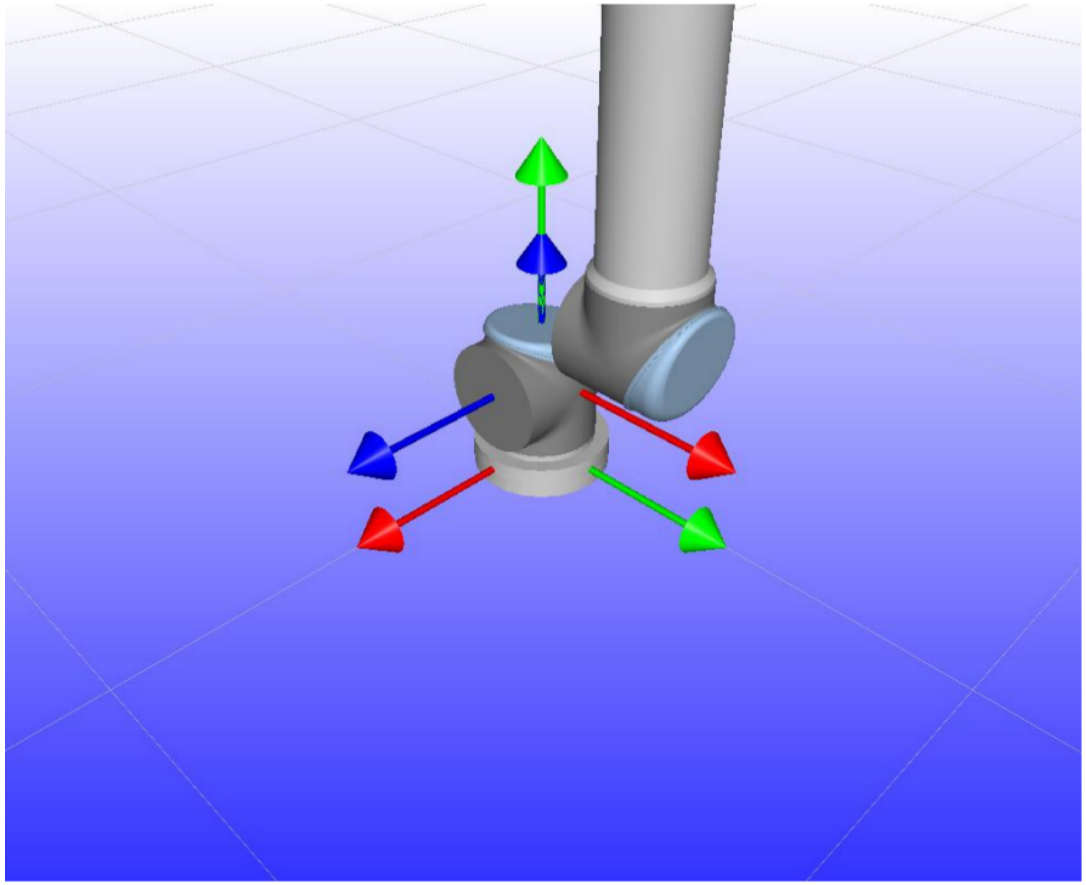


Figure: Drawable1: rotate drawing ($R = 270°$)

Programming Exercise 2

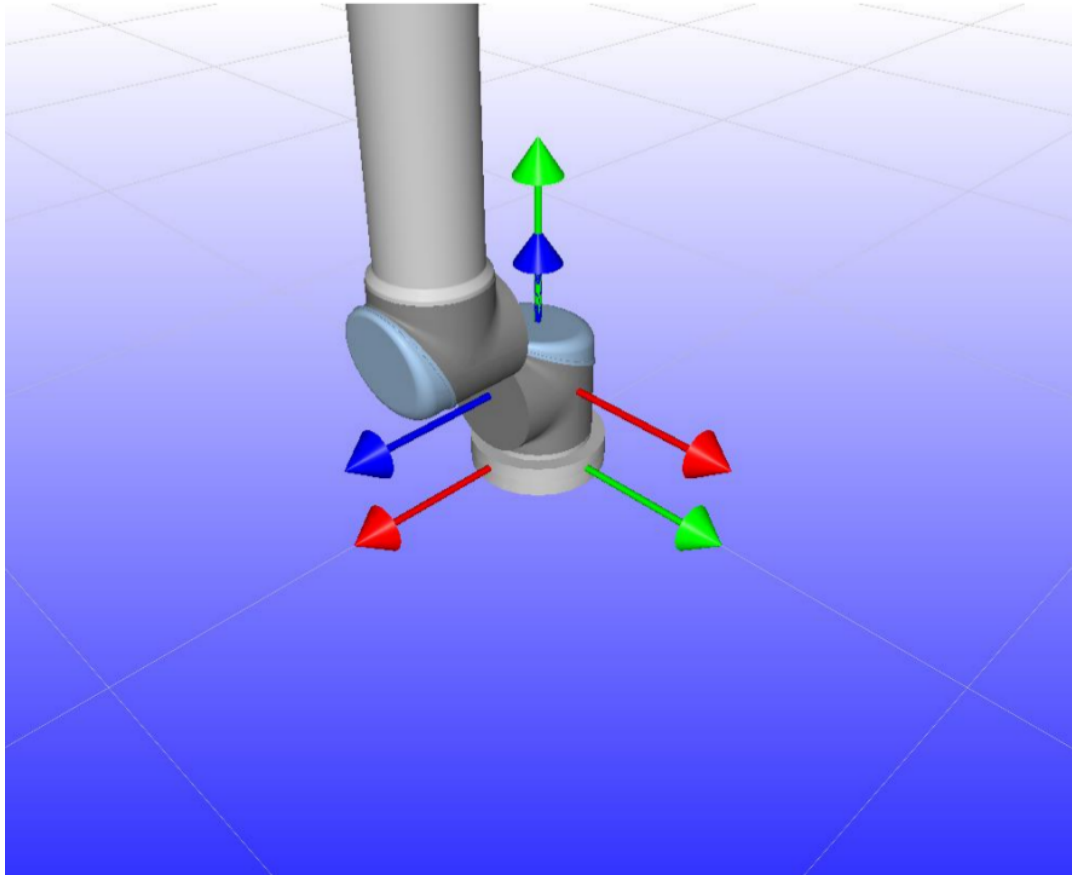


Figure: Drawable1: rotate drawing ($Y = 90°$)

Programming Exercise 2

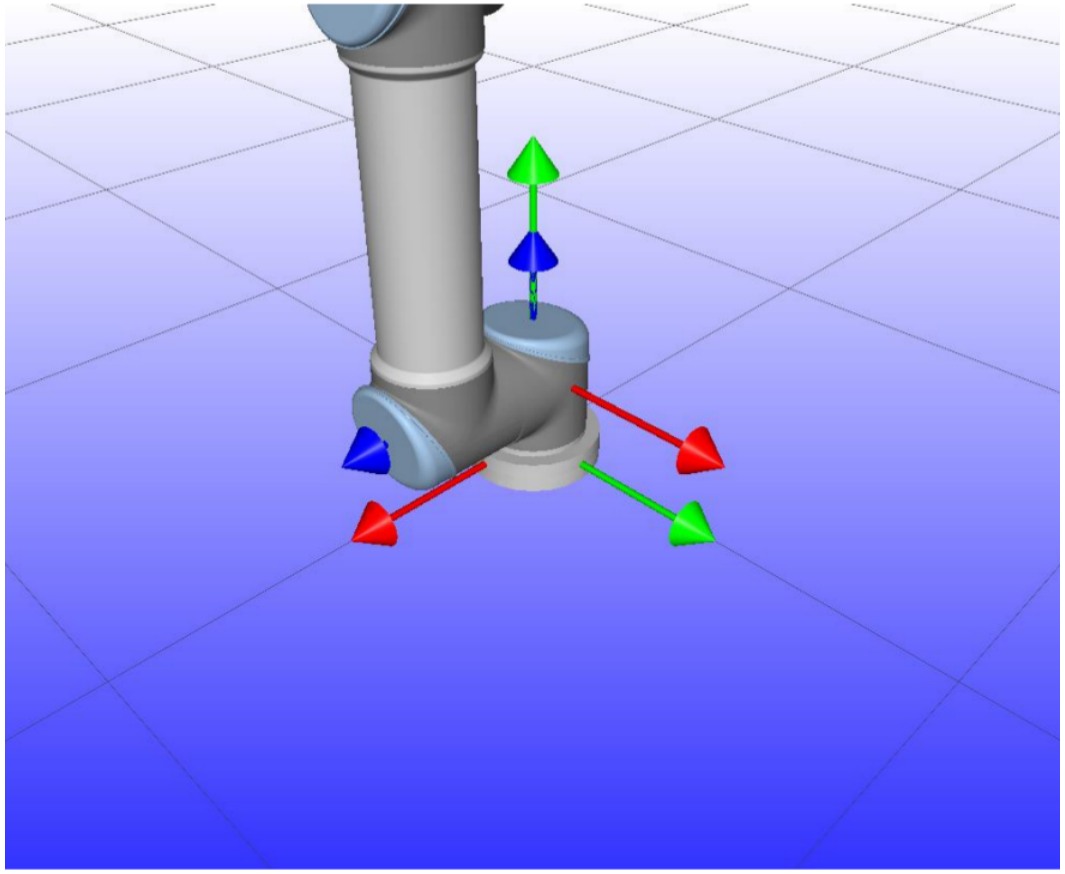


Figure: Drawable1: move drawing ($y = -0.08920$)

Programming Exercise 2

- Joint1 in place.
- ```
  <Joint name="Joint1" type="Revolute">
  <RPY> 90 0 90 </RPY> <Pos> 0 0 0.0892 </Pos>
  </Joint>
  <Drawable name="Joint1Geo" refframe="Joint1">
  <RPY> 270 0 90 </RPY> <Pos> 0 -0.0892 0</Pos>
  <Polytope file="Geometry/joint1"/>
  </Drawable>
  <Q name="Home">0 0</Q>
  ```
  ```

Tips

- Be systematic in your approach. Either:
  - Rotations before positions
  - Positions before rotations
- Remember to make the home Q vector (end of XML) the right size
- Use the diagram from the datasheet for:
  - Dimensions of the robot
  - Position/Orientation of frames
- There are small misalignments in the drawables. Ignore these!

References

L. P. Ellekilde and J. A. Jorgensen (2010)

RobWork: A Flexible Toolbox for Robotics Research and Education

ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), 1 – 7.