Mandatory Exercise 2 and LUA

Guðmundur Geir Gunnarsson

University of Southern Denmark

gunu@mmmi.sdu.dk

September 27, 2017

Overview

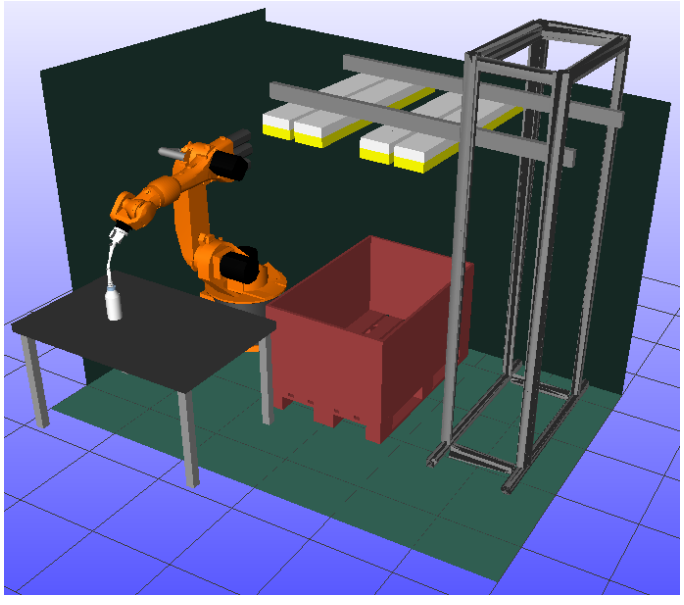
Pathplanning

Lua

About the exercise

Hand-in date

Pathplanning

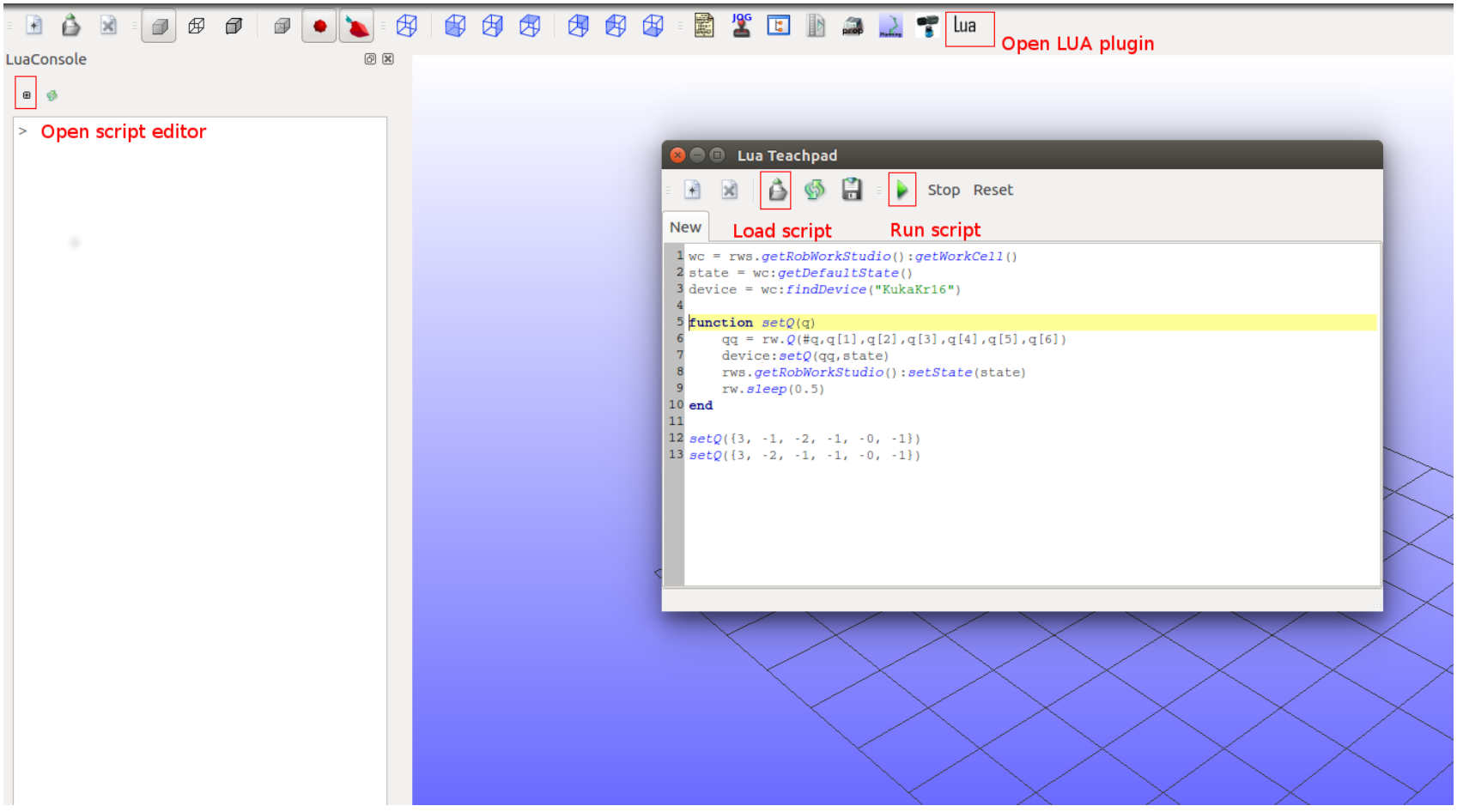


- To solve the exercise
  - Perform path planning and calculate statistics
  - Extend `pathplanning.cpp`
  - Use the workcell *Kr16WallWorkCell*
  - Create a function that exports the configuration path to LUA
  - Run the LUA script in RobWorkStudio to visualize the path

Lua

- Lua is a scripting language that can be used in Robworkstudio to visualize movement
- You can attach objects to robot frames
- Kinematic movements (discrete steps)

Using Lua


Lua
Open LUA plugin
LuaConsole
> Open script editor
Lua Teachpad
Stop
Reset
New
Load script
Run script
1 wc = rws.getRobWorkStudio():getWorkCell()
2 state = wc:getDefaultState()
3 device = wc:findDevice("KukaKr16")
4
5 function setQ(q)
6 qq = rw.Q(#q,q[1],q[2],q[3],q[4],q[5],q[6])
7 device:setQ(qq,state)
8 rws.getRobWorkStudio():setState(state)
9 rw.sleep(0.5)
10 end
11
12 setQ({3, -1, -2, -1, -0, -1})
13 setQ({3, -2, -1, -1, -0, -1})

Basic Script

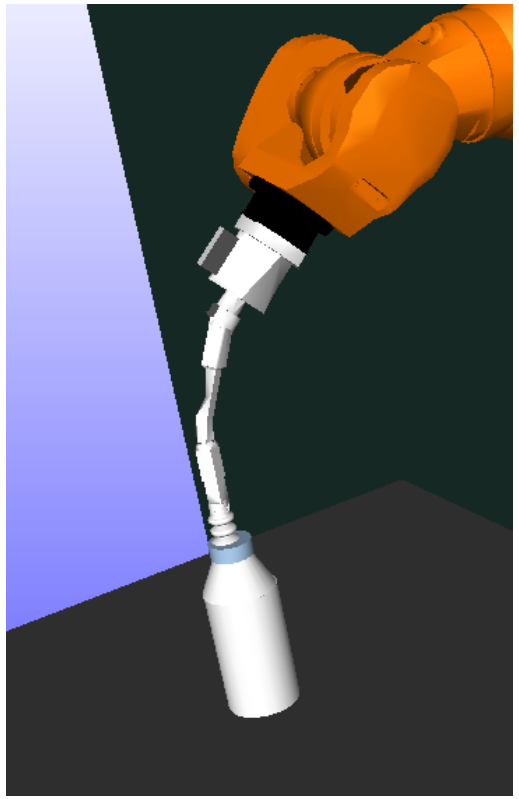
```
wc = rws.getRobWorkStudio():getWorkCell()
state = wc:getDefaultState()
device = wc:findDevice(``KukaKr16'')

function setQ(q)
      qq = rw.Q(#q, q[1], q[2], q[3], q[4], q[5], q[6])
      device:setQ(qq, state)
      rws.getRobWorkStudio():setState(state)
      rw.sleep(0.5)
end

setQ({3, -1, -2, -1, -0, -1})
setQ({3, -2, -1, -1, -0, -1})
```

Grasping the Bottle

- Grasping the bottle in C++ and Lua:
  - See `kinematics::gripFrame()` in the API
  - Remember to set the *state* (`rw::kinematics::state`)
- Remember
  - The `PlannerConstraint` needs to be constructed with a state
  - If you change the state after you construct the `PlannerConstraint`, it needs to be reconstructed along with the `QToQPlanner`



Repeatability of Results

- The RRT planner is probabilistic (uses a random generator)
- In order to get different results each time you need to set the RNG seed with:
  - `rw::math::Math::seed()`, uses the current date, or
  - `rw::math::Math::seed(int)`, uses the int as a seed
- The seed should be set at the start of your program

Hand-in date

- Hand-in date is 23rd October at 12.00
- Send email to me (`gunu@mmmi.sdu.dk`) and Henrik (`hgp@mmmi.sdu.dk`) with the title "RobMand2"
- One hand in per group
- Zip your files
- Hand in: C++ code, Lua script, and a small report about your results