sim=require'sim'

simIK=require'simIK'

function hopThroughConfigs(path,joints,reverse,dynModel)

local lb=sim.setStepping(true)

local s=1

local g=#path/6

local incr=1

if reverse then

s=#path/6

g=1

incr=-1

end

for i=s,g,incr do

if dynModel then

for j=1,#joints,1 do

sim.setJointTargetPosition(joints[j],path[(i-1)\*6+j])

end

else

for j=1,#joints,1 do

sim.setJointPosition(joints[j],path[(i-1)\*6+j])

end

end

sim.step()

end

sim.setStepping(lb)

end

function sysCall\_thread()

local simBase=sim.getObject('..')

local simTip=sim.getObject('../tip')

local simGoal=sim.getObject('/goalPose')

local simJoints={}

for i=1,6,1 do

simJoints[i]=sim.getObject('../joint',{index=i-1})

end

sim.step() -- make sure we have skipped the first simulation step,

-- otherwise following cmd won't reflect reality

local dynModel=sim.isDynamicallyEnabled(simJoints[1])

-- Prepare an ik group, using the convenience function 'simIK.addElementFromScene':

local ikEnv=simIK.createEnvironment()

local ikGroup=simIK.createGroup(ikEnv)

local ikElement,simToIkMap=simIK.addElementFromScene(ikEnv,ikGroup,simBase,simTip,simGoal,simIK.constraint\_pose)

-- Retrieve some handles of objects created in the IK environment:

local ikTip=simToIkMap[simTip]

local ikJoints={}

for i=1,#simJoints,1 do

ikJoints[i]=simToIkMap[simJoints[i]]

end

-- Generate a path:

local path=simIK.generatePath(ikEnv,ikGroup,ikJoints,ikTip,300)

simIK.eraseEnvironment(ikEnv)

-- Hop through the path configurations:

while true do

hopThroughConfigs(path,simJoints,false,dynModel)

hopThroughConfigs(path,simJoints,true,dynModel)

end

end