University of California San Diego

Course: MAE 204 - Robotics

LAB 2 - Tower of Hanoi with ROS

February 13, 2023

Authors	*Student ID*
Aashish Joshi	A59020875
Mihir Kulkarni	A59018127
Nandan Seshadri	A59018150
Surya Prakash Thoguluva Kumaran Babu	A59017860

Overview

Universal Robot 3e is a table top robot for assembly tasks. The UR3e was programmed and locally controlled using a teach pendant to perform a basic Tower of Hanoi task. Following the rules of the Tower of Hanoi, the UR3e was successfully programmed. This experiment was performed in lab 1 and the observations are now being used in a simulated environment.

The detailed instructions helped us navigate through the new environment and successfully simulated the Tower of Hanoi in Ubuntu. A brief overview of the steps are as follows:

1. Installing and configuring the virtual machine
2. Linux commands to build the workspace
3. Using ROS packages to run nodes
4. Provided joint angles
5. Simulated the task

Video Link

Lab 2

Results

The robot performed the Tower of Hanoi task with 2 blocks. UR3e robot goes through various configurations for the tasks of going to the block coordinates, closing and opening the gripper and stacking the blocks on top of each other.
With a good understanding of ROS and Ubuntu interface, the experiment was simulated and completed successfully.

Appendix

```
import rospy
import numpy as np

from ur_control.arm import Arm
from ur_control import transformations

def move_joints(joint_angles):
    print("Moving to waypoint...")
          arm.set_joint_positions(position=joint_angles,  velocities=None,
accelerations=None, wait=True, t=2.0)

def open_gripper(link_name=None):
    print("Opening gripper...")
    arm.gripper.open()
    if link_name!=None:
        arm.gripper.release(link_name)

def close_gripper(link_name=None):
    print("Closing gripper...")
    if link_name==None:
        arm.gripper.close()
    else:
        arm.gripper.command(0.03)
        arm.gripper.grab(link_name)


#Joint angles
def lab2_task():
      standby=np.array([-0.20835858980287725,           -1.44597980499778,
1.4588902632342737,        -1.570404260801599,        -1.5696094671832483,
2.9373891353607178])
      Waypoint_1=np.array([0.04942440986633301,         -0.7750004094890137,
1.127181355153219,         -1.909250875512594,         -1.5668476263629358,
3.196812942504883])
      Waypoint_2=np.array([0.03199164941906929,         -0.7101233762553711,
1.2127340475665491,         -2.0596934757628382,          -1.56736928621401,
3.1797032356262207])
      Waypoint_3=np.array([0.03172876685857773,         -0.8428984445384522,
1.032461945210592,         -1.746650835076803,         -1.5669220129596155,
3.178842067718506])
      Waypoint_4=np.array([-0.6234691778766077,         -0.7923308175853272,
1.0529630819903772,        -1.8209248981871546,        -1.5758398214923304,
2.523887872695923])
      Waypoint_5=np.array([-0.6044567267047327,         -0.6390751761249085,
1.2172873655902308,        -2.138510366479391,         -1.5761101881610315,
2.543503999710083])
      Waypoint_18=np.array([-0.7145336310016077,        -1.0497335952571412,
1.1436675230609339,        -1.6551348171629847,         -1.576355282460348,
2.432103395462036])
      Waypoint_6=np.array([0.030917322263121605,        -0.8421853345683594,
1.0379584471331995,        -1.7528664074339808,         -1.566887680684225,
3.1779916286468506])
      Waypoint_7=np.array([0.03122328780591488,         -0.6412031215480347,
1.2388399283038538,        -2.154806753198141,         -1.5676010290728968,
3.179128646850586])
       Waypoint_8=np.array([0.031012821942567825,       -0.8112590473941346,
1.1121123472796839,        -1.8579722843565882,         -1.567014519368307,
3.1783199310302734])
      Waypoint_9=np.array([-0.3318989912616175,           -0.90915401399646,
1.287851635609762,        -1.9368764362730921,         -1.5722387472735804,
2.815415143966675])
      Waypoint_10=np.array([-0.29775792757143194,      -0.7311768692782898,
1.4079578558551233,        -2.234641214410299,         -1.5723360220538538,
2.850123405456543])
```

```
      Waypoint_11=np.array([-0.2981069723712366,      -0.9561102551272889,
1.1995042006122034,       -1.8012448749937953,       -1.5715306440936487,
2.8489513397216797])
      Waypoint_12=np.array([-0.6108997503863733,      -0.832878128891327,
0.9872720877276819,       -1.7145163021483363,       -1.5755165258990687,
2.536292314529419])
      Waypoint_13=np.array([-0.6105559507953089,      -0.6309222143939515,
1.2139790693866175,       -2.143144746819967,        -1.5762847105609339,
2.5373995304107666])
      Waypoint_14=np.array([-0.6107195059405726,      -0.7887136501124878,
1.100520435963766,        -1.8718720875182093,       -1.5757601896869105,
2.5366880893707275])
      Waypoint_15=np.array([-0.30040103593935186,     -0.9009845417789002,
1.30273944536318,         -1.9596172771849574,       -1.5718353430377405,
2.846829414367676])
      Waypoint_16=np.array([-0.2962277571307581,      -0.8088823121837159,
1.3860872427569788,       -2.135127683679098,        -1.5720813910113733,
2.8514187335968018])
      Waypoint_17=np.array([-0.2966845671283167,      -0.9827328485301514,
1.1080191771136683,       -1.683176179925436,        -1.5714286009417933,
2.850154399871826])

#Program sequence
    move_joints(standby)
    open_gripper(link_name="cube1::link")
    move_joints(Waypoint_1)
    move_joints(Waypoint_2)
    close_gripper(link_name="cube1::link")
    move_joints(Waypoint_3)
    move_joints(Waypoint_4)
    move_joints(Waypoint_5)
    open_gripper(link_name="cube1::link")
    move_joints(Waypoint_18)
    move_joints(Waypoint_6)
    move_joints(Waypoint_7)
    close_gripper(link_name="cube2::link")
    move_joints(Waypoint_8)
    move_joints(Waypoint_9)
    move_joints(Waypoint_10)
    open_gripper(link_name="cube2::link")
    move_joints(Waypoint_11)
    move_joints(Waypoint_12)
    move_joints(Waypoint_13)
    close_gripper(link_name="cube1::link")
    move_joints(Waypoint_14)
    move_joints(Waypoint_15)
    move_joints(Waypoint_16)
    open_gripper(link_name="cube1::link")
    move_joints(Waypoint_17)


def main():
    rospy.init_node("lab2", log_level=rospy.INFO)

    ns = ''
    joints_prefix = None
    robot_urdf = "ur3e"
    rospackage = None
    tcp_link = None
    use_gripper = True

    global arm
    arm = Arm(ft_sensor=False,
              gripper=use_gripper, namespace=ns,
              joint_names_prefix=joints_prefix,
              robot_urdf=robot_urdf, robot_urdf_package=rospackage,
              ee_link=tcp_link)

    lab2_task()
    print("Done.")


if __name__ == '__main__':
    main()
```