Collaborative Robotic System for Cup Stacking

A-4조
정승환 장준하 윤민식

OUTLINE

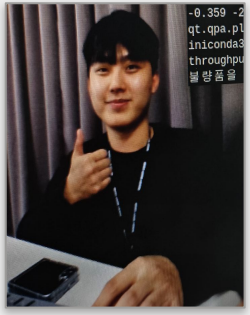
1. Code review
2. Result
3. Conclusion & Future Work

Introduction

Members

윤민식

시골 청년

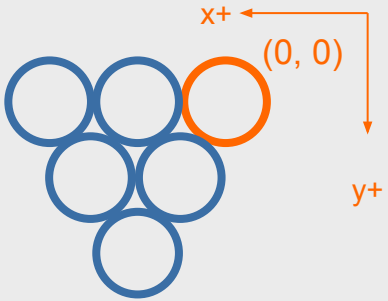


정승환

장준하

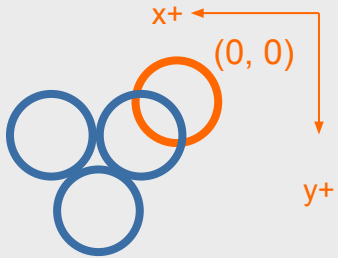
Code review

Stacking algorithm



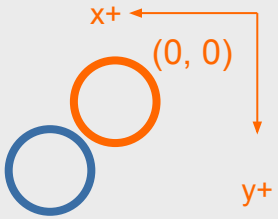
1st floor

(2, 0) (1, 0) (0, 0)
(1.5, 1) (0.5, 1)
(1, 2)



2nd floor

(1.5, 0.5) (0.5, 0.5)
(1, 1.5)

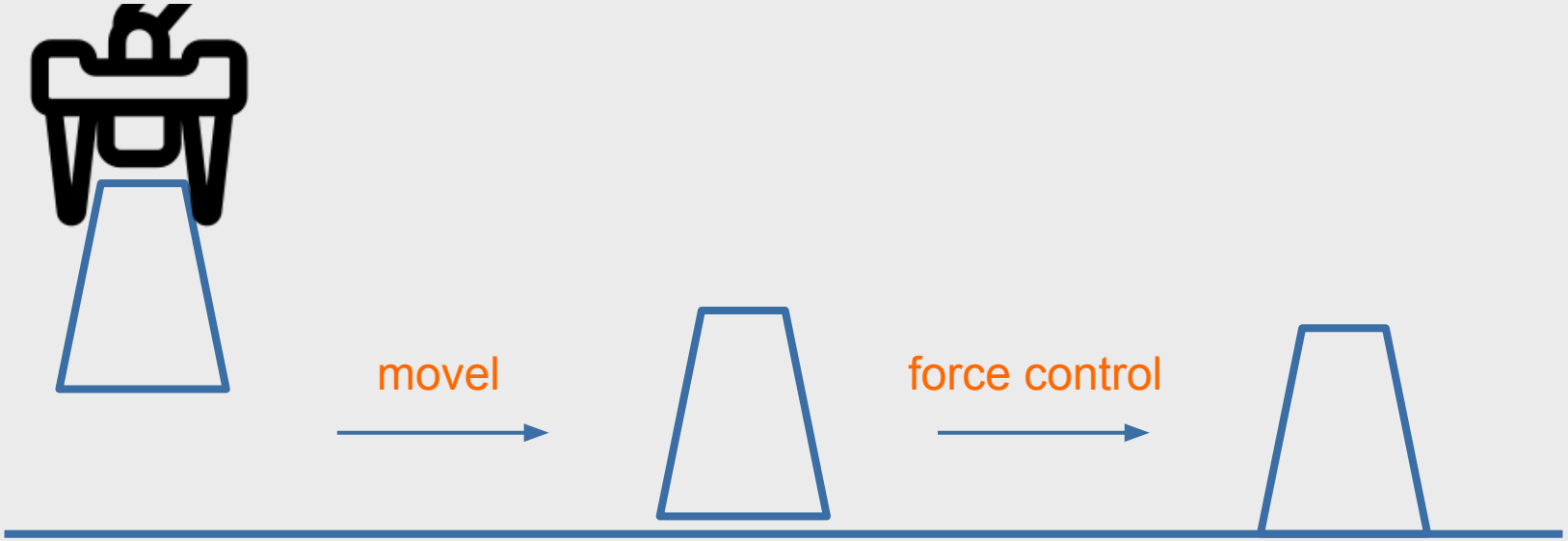


3rd floor

(1, 1)

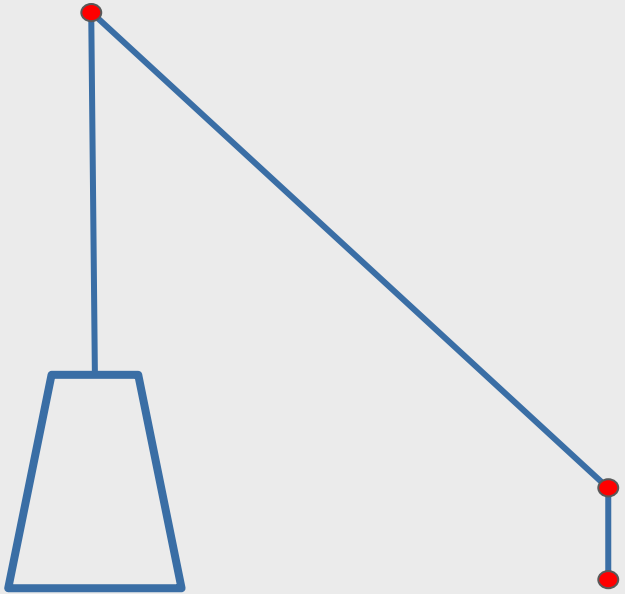
Code review

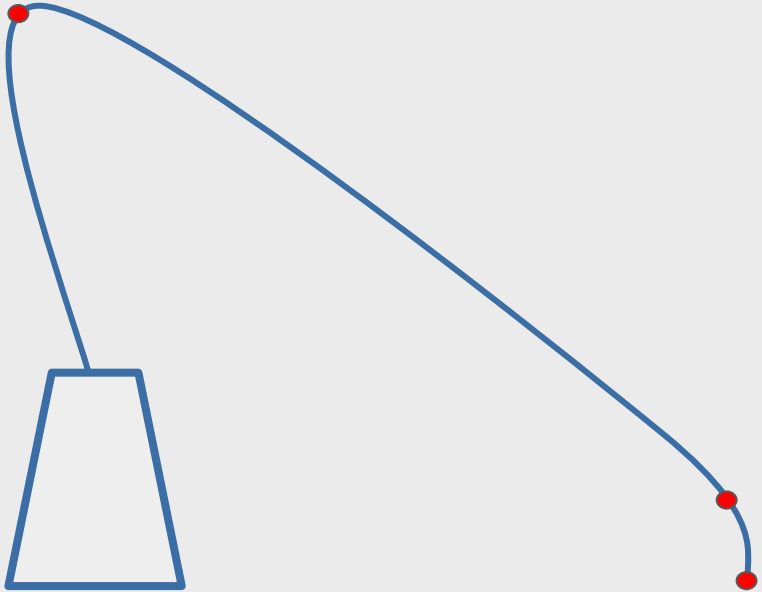
force control



Code review

Spline path





Code review

```python
import rclpy
import DR_init

# for single robot
ROBOT_ID = "dsr01"
ROBOT_MODEL = "m0609"
VELOCITY, ACC = 700, 700

# Control CONSTANT
CUP_DIAMETER_X = 80              # CUP x좌표 가중치
CUP_DIAMETER_Y = 73              # CUP y좌표 가중치
HEIGHT_PLACE_OFFEST = 30         # CUP 이동시 충돌 방지를 위한 OFFSET
HEIGHT_TARGET_OFFSET = 20        # CUP 회전시 충돌 방지를 위한 OFFSET
MAX_FORCE = 8

DR_init.__dsr__id = ROBOT_ID
DR_init.__dsr__model = ROBOT_MODEL

OFF, ON = 0, 1

spline_list = [] # movesx를 위한 list
```

```python
def main(args=None):
    rclpy.init(args=args)
    node = rclpy.create_node("cup_stacking", namespace=ROBOT_ID)

    DR_init.__dsr__node = node

    try:
        from DSR_ROBOT2 import (
            release_compliance_ctrl,
            check_force_condition,
            task_compliance_ctrl,
            set_digital_output,
            set_desired_force,
            get_digital_input,
            get_current_posx,
            DR_FC_MOD_REL,
            DR_AXIS_Z,
            set_tool,
            set_tcp,
            DR_BASE,
            movesx,
            movej,
            movel,
            wait,
        )

        from DR_common2 import posx

    except ImportError as e:
        print(f"Error importing DSR_ROBOT2 : {e}")
        return
```

Code review

```
    # init pose
    JReady = [0, 0, 90, 0, 90, 0]


    # 컵 쌓여있는 위치
    world_cup_pose = posx([420.50, -209.33, 246.11, 0.0, 180.0, 0.0])
    # 뒤집는 컵 위치
    last_cup_grip_pose = posx([421.75, -179.06, 100.73, 88.28, 90.0, 90.0])


    # 시작 좌표
    place_start_pose = posx([561.52, 221.63, 85.13, 0.0, 180.0, 0.0])
    fourth_floor_start_pose = posx([561.52, 221.63, 306.26, 95.25, 90.0, -90.0])

    floor_list = [
        [(0, 0), (1.0, 0), (2.0, 0), (0.5, 1.0), (1.5, 1.0), (1.0, 2.0)],
        [(0.5, 0.5), (1.5, 0.5), (1.0, 1.5)],
        [(1.0,1.0)]
    ]

    fourth_floor_list = (1.0,0.6)

    floor_z = [
        85.13,
        178.1,
        272.07
    ]

    # 특이점 탈출 좌표
    out_of_sing = ([-44.23, 26.22, 89.20, 63.99, 98.94, -28.57])

    # 컴플라이언스
    compliance_ctrl_stx = [500, 500, 500, 100, 100, 100]
    fd_z_minus = [0, 0, -80, 0, 0, 0]
    fd_dir_z = [0, 0, 1, 0, 0, 0]
```

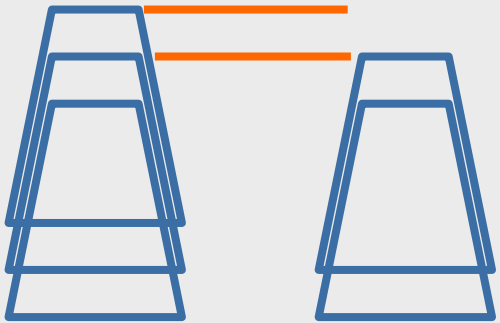
```
def wait_digital_input(sig_num):
    while not get_digital_input(sig_num):
        wait(0.5)
        print("Wait for digital input")
        pass

def release():
    set_digital_output(2, ON)
    set_digital_output(1, OFF)
    wait_digital_input(2)

def grip():
    release()
    set_digital_output(1, ON)
    set_digital_output(2, OFF)
    wait_digital_input(1)
```

Code review

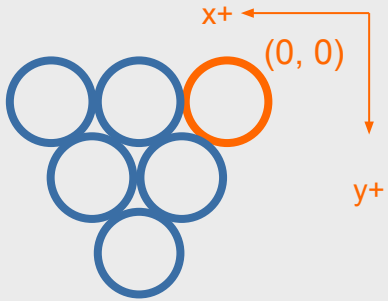
world_cup()



```
103  def world_cup():
104
105      movel(world_cup_pose, vel = VELOCITY, acc = ACC, ref = DR_BASE)
106
107      grip()
108
109      task_compliance_ctrl(stx=compliance_ctrl_stx)
110      set_desired_force(fd=fd_z_minus, dir=fd_dir_z, mod=DR_FC_MOD_REL)
111      while not check_force_condition(DR_AXIS_Z, max=10):
112          pass
113
114      pos = get_current_posx()[0]
115
116      release_compliance_ctrl()
117
118      pos[2] += 10
119
120      movel(pos, vel = VELOCITY, acc = ACC, ref = DR_BASE)
121
122      world_cup_z = get_current_posx()[0][2]
123      world_cup_pose[2] = world_cup_z
124
125      release()
126
127      pos[2] -= 18
128
129      movel(pos, vel = VELOCITY, acc = ACC, ref = DR_BASE)
130
131      grip()
132
133      pos_2 = get_current_posx()[0]
134
135      pos_2[2] += 125 # 125
136
137      global spline_list
138
139      spline_list.append(posx(pos_2))
```

Code review

first_floor()

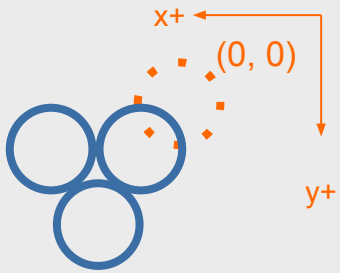



```
floor_list = [
    [(0, 0), (1.0, 0), (2.0, 0), (0.5, 1.0), (1.5, 1.0), (1.0, 2.0)],
    [(1.0,1.0)]
]
```

```
142     def stacking(start_pose,pos_z,coordinate):
143         global spline_list
144
145         for xy in coordinate:
146             world_cup()
147
148             pose = start_pose.copy()
149
150             pose[2] = pos_z
151             pose[0] -= CUP_DIAMETER_X * xy[0]
152             pose[1] -= CUP_DIAMETER_Y * xy[1]
153             pose[2] += HEIGHT_PLACE_OFFEST
154
155             spline_list.append(posx(pose))
156             pose[2] -= HEIGHT_PLACE_OFFEST
157             spline_list.append(posx(pose))
158
159             movesx(spline_list, vel = VELOCITY, acc = ACC, ref = DR_BASE)
160
161             spline_list = []
162
163             task_compliance_ctrl(stx=compliance_ctrl_stx)
164             set_desired_force(fd=fd_z_minus, dir=fd_dir_z, mod=DR_FC_MOD_REL)
165             while not check_force_condition(DR_AXIS_Z, max=MAX_FORCE):
166                 pass
167
168             release_compliance_ctrl()
169
170             release()
```

Code review

second_floor()

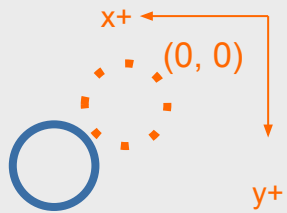



```
floor_list = [
    [(0, 0), (1.0, 0), (2.0, 0), (0.5, 1.0), (1.5, 1.0), (1.0, 2.0)],
    [(0.5, 0.5), (1.5, 0.5), (1.0, 1.5)],
    [(1.0,1.0)]
]
```

```
142     def stacking(start_pose,pos_z,coordinate):
143         global spline_list
144 
145         for xy in coordinate:
146             world_cup()
147 
148             pose = start_pose.copy()
149 
150             pose[2] = pos_z
151             pose[0] -= CUP_DIAMETER_X * xy[0]
152             pose[1] -= CUP_DIAMETER_Y * xy[1]
153             pose[2] += HEIGHT_PLACE_OFFEST
154 
155             spline_list.append(posx(pose))
156             pose[2] -= HEIGHT_PLACE_OFFEST
157             spline_list.append(posx(pose))
158 
159             movesx(spline_list, vel = VELOCITY, acc = ACC, ref = DR_BASE)
160 
161             spline_list = []
162 
163             task_compliance_ctrl(stx=compliance_ctrl_stx)
164             set_desired_force(fd=fd_z_minus, dir=fd_dir_z, mod=DR_FC_MOD_REL)
165             while not check_force_condition(DR_AXIS_Z, max=MAX_FORCE):
166                 pass
167 
168             release_compliance_ctrl()
169 
170             release()
```

Code review

third_floor()




```
floor_list = [
    [(0, 0), (1.0, 0), (2.0, 0), (0.5, 1.0), (1.5, 1.0), (1.0, 2.0)],
    [(0.5, 0.5), (1.5, 0.5), (1.0, 1.5)],
    [(1.0,1.0)]
]
```

```
142     def stacking(start_pose,pos_z,coordinate):
143         global spline_list
144 
145         for xy in coordinate:
146             world_cup()
147 
148             pose = start_pose.copy()
149 
150             pose[2] = pos_z
151             pose[0] -= CUP_DIAMETER_X * xy[0]
152             pose[1] -= CUP_DIAMETER_Y * xy[1]
153             pose[2] += HEIGHT_PLACE_OFFEST
154 
155             spline_list.append(posx(pose))
156             pose[2] -= HEIGHT_PLACE_OFFEST
157             spline_list.append(posx(pose))
158 
159             movesx(spline_list, vel = VELOCITY, acc = ACC, ref = DR_BASE)
160 
161             spline_list = []
162 
163             task_compliance_ctrl(stx=compliance_ctrl_stx)
164             set_desired_force(fd=fd_z_minus, dir=fd_dir_z, mod=DR_FC_MOD_REL)
165             while not check_force_condition(DR_AXIS_Z, max=MAX_FORCE):
166                 pass
167 
168             release_compliance_ctrl()
169 
170             release()
```

Code review

fourth_floor()

singularity avoidance

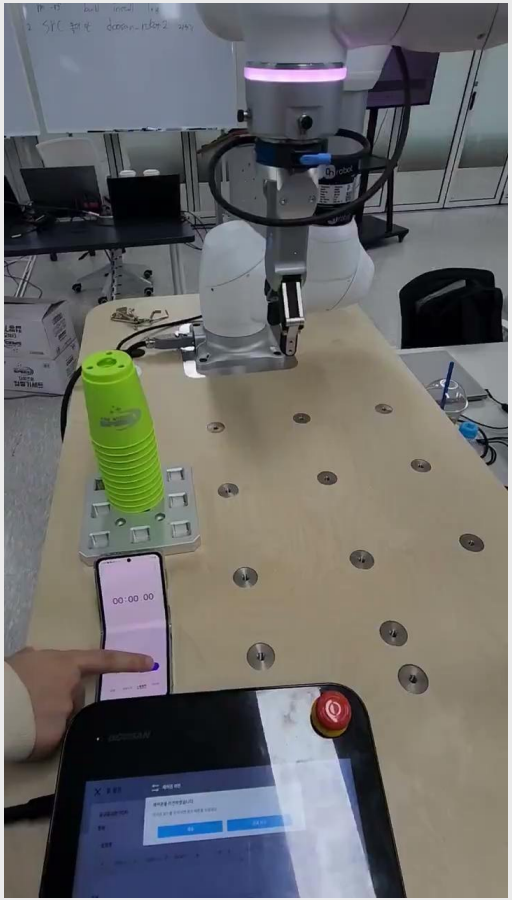
```
246     def fourth__floor(start_pose):
247             #fourth floor
248             release()
249
250             pose = start_pose.copy()
251
252             pose[0] -= CUP_DIAMETER_X * fourth_floor_list[0]
253
254             pose[1] -= CUP_DIAMETER_Y * fourth_floor_list[1]
255
256             last_cup_grip_pose[2] += HEIGHT_PLACE_OFFEST
257
258             movel(last_cup_grip_pose, vel = 100, acc = 100)
259
260             last_cup_grip_pose[2] -= HEIGHT_PLACE_OFFEST
261
262             movel(last_cup_grip_pose, vel = VELOCITY, acc = ACC)
263
264             grip()
265
266             last_cup_grip_pose[2] += HEIGHT_PLACE_OFFEST
267
268             movel(last_cup_grip_pose, vel = VELOCITY - 400, acc = ACC - 400, ref = DR_BASE)
269
270             movej(out_of_sing, vel = 80, acc = 80)
271
272             pose[2] += 20
273
274             movel(pose, vel = VELOCITY, acc = ACC, ref = DR_BASE)
275
276             pose[2] -= 20
277
278             movel(pose, vel = VELOCITY, acc = ACC, ref = DR_BASE)
279
280             task_compliance_ctrl(stx=compliance_ctrl_stx)
281             set_desired_force(fd=fd_z_minus, dir=fd_dir_z, mod=DR_FC_MOD_REL)
282             while not check_force_condition(DR_AXIS_Z, max=MAX_FORCE):
283                 pass
284
285             release_compliance_ctrl()
286
287             release()
```

Code review

```
207     set_tool("Tool Weight")
208     set_tcp("GripperSA_v1_test_1")
209
210     movej(JReady, vel=30, acc=30)           # 초기 위치로
211
212     for coor,pos_z in zip(floor_list,floor_z):
213         stacking(place_start_pose,pos_z,coor)
214
215     fourth__floor(fourth_floor_start_pose)
216
217     rclpy.shutdown()
218
219
220 if __name__ == "__main__":
221     main()
```

Result

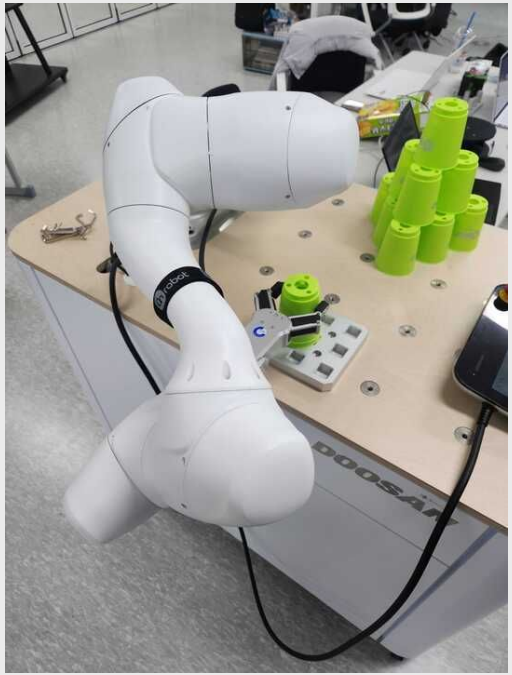
result video



소요 시간
1:48:8

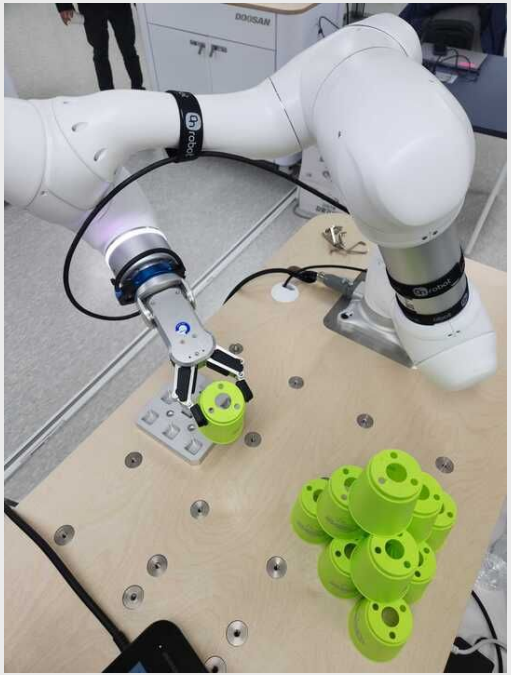
Conclusion & Future Work

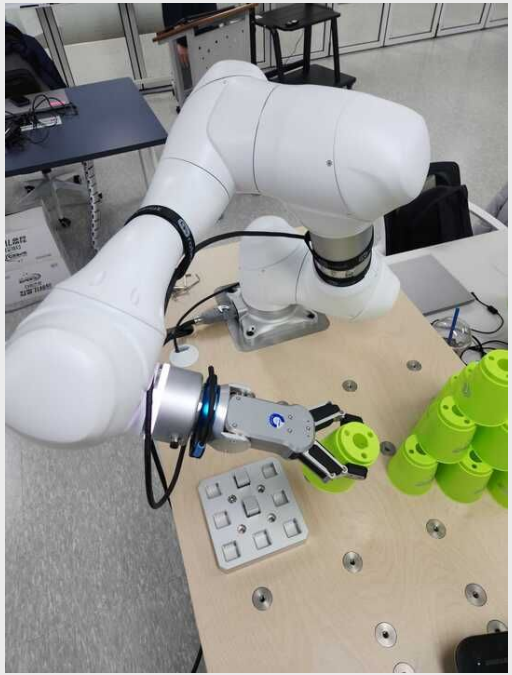
Conclusion



Conclusion & Future Work

Conclusion





Conclusion & Future Work

Future Work

- 로봇팔의 최적의 **Joint** 각도를 구현하여 특이점 회피 및 안정적인 동작 구현
- 불필요한 움직임을 제거하여 효율적인 **Path Planning**
- 목표 물체의 정확한 위치 및 움직임을 최소화
- **OFFSET**값을 조절하여 **force control** 소요시간 최소화

Q&A

감사합니다