
협동 로봇을 통한 피라미드 컵 쌓기

협동 2

김서광 , 박성재 , 신정훈 , 최민준

Contents

1

팀 소개 & 목표

3

퍼포먼스

2

코드 리뷰

4

프로젝트를
마치며

1. 팀 소개 & 목표

도파민 중독자들

김서광 (중독자 1)

박성재 (중독자 2)

- 목표를 달성하며 뿔어져 나오는 도파민을 사랑하는 팀원들
- 사다리타기로 발표자를 정하는 소소한 도파민도 놓치지 않는 팀

신정훈 (중독자 3)

최민준 (중독자 4)

1. 팀 소개 & 목표



협동 로봇을 활용하여 6 3 1 1(뒤집기) 구조의 피라미드 컵 쌓기

2. 코드 리뷰

1 ~ 3 단 컵 쌓기

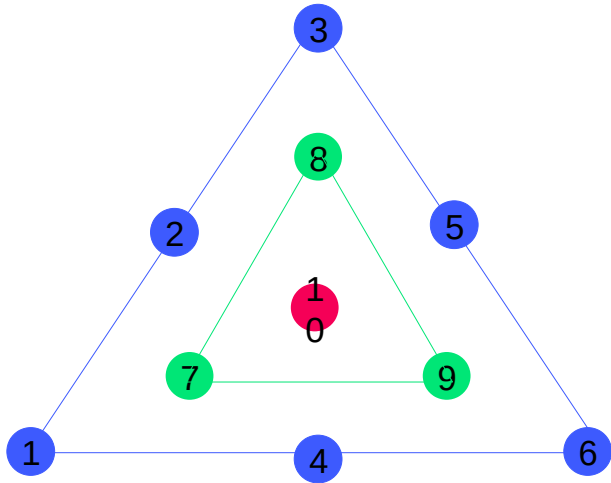
```
for i in range(9):  
    add=trans(point[0], delta[i], DR_BASE, DR_BASE)  
    add = add.tolist()  
    point.append(add)
```

- trans(기준 좌표 , 변화량 , 기준 좌표계 , 변환할 좌표계) 를 통하여 add 좌표를 생성
- trans 는 numpy.array 타입을 반환하기에 movei, movej 등에 활용하게 위해서는 tolist() 함수로 list 타입으로 변환해주어야함
- point 리스트에 변환한 좌표들을 추가

2. 코드 리뷰

1~3 단 컵 쌓기

```
root_3=3**(1/2)
delta = [[38*root_3, -38, 0, 0, 0, 0],
         [76*root_3, -76, 0, 0, 0, 0],
         [0, -76, 0, 0, 0, 0],
         [38*root_3, -114, 0, 0, 0, 0],
         [0, -152, 0, 0, 0, 0],
         [38*root_3/3, -38, 96, 0, 0, 0],
         [38*root_3*4/3, -76, 96, 0, 0, 0],
         [38*root_3/3, -114, 96, 0, 0, 0],
         [76*root_3/3, -76, 192, 0, 0, 0]]
```



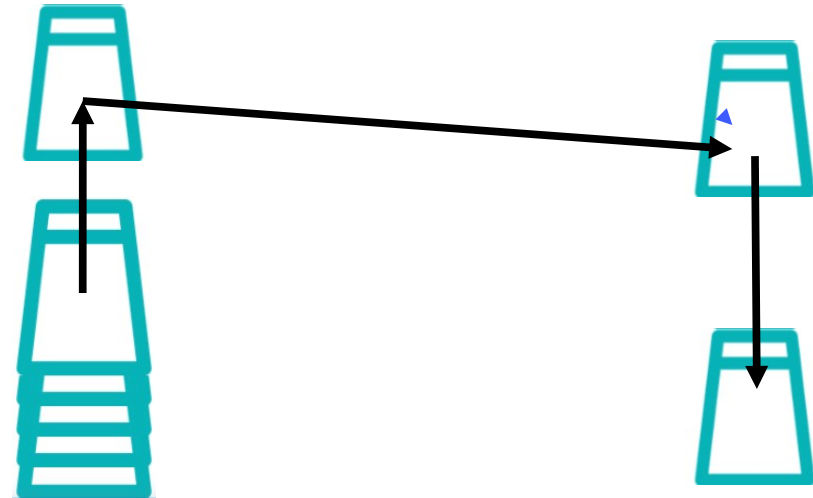
- 기준 (point[0]): 왼쪽 아래 컵
- 컵 spec
 - 지름 : 76mm
 - 높이 : 95mm
- 1,2,3 층 : 파랑 초록 빨강
- 위층 컵은 주변 3 개의 아래층 컵의 무게중심에 위치
- 무게 중심의 특징과 정삼각형의 특징을 이용하여 각 컵들의 상대 위치를 계산

2. 코드 리뷰

1~3 단 컵 쌓기

```
def pick_up_put_down(n):  
    real_grap=grap_x[:]  
    real_grap[2]-=11.2*n  
    real_grap_up=real_grap[:]  
    real_grap_up[2]+=120  
    put_up=point[n][:]  
    put_up[2]+=100  
  
    move1(real_grap,vel=VELOCITY,acc=ACC)  
    grip()  
    movesx([posx(real_grap_up),posx(put_up),posx(point[n])],vel=VELOCITY,acc=ACC)  
    release()
```

- 컵이 쌓여있는 위치에 바로 접근 하여 잡은 후 ,
- movesx 에 리스트로 갈 곳을 입력하여 컵을 쌓는 것을 한번에 부드럽게 실행



2. 코드 리뷰

movesx()

3.3.6 movesx()

기능

로봇이 현재 위치에서 pos_list로 입력된 작업공간(task space)의 경유점들을 거쳐 목표위치(pos_list의 마지막 경유점)까지 연결되는 스플라인 곡선경로를 따라 이동합니다.

입력된 속도/가속도는 경로 중 최대 속도/가속도이며 정속모션 옵션을 선택할 경우 조건에 따라 입력한 속도로 정속도의 모션을 수행합니다.

인수

인수명	자료형	기본값	설명
pos_list	list (posx)	-	posx list
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])		
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])		
time (t)	float	None	도달 시간 [sec]
ref	int	None	reference coordinate <ul style="list-style-type: none">DR_BASE: base coordinateDR_WORLD: world coordinateDR_TOOL: tool coordinateuser coordinate: 사용자 정의
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none">DR_MV_MOD_ABS: 절대DR_MV_MOD_REL: 상대

인수명	자료형	기본값	설명
vel_opt	int	DR_MVS_VEL_NONE	속도 옵션 <ul style="list-style-type: none">DR_MVS_VEL_NONE: 없음DR_MVS_VEL_CONST: 등속

① 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우 _global_velx가 적용됩니다. (_global_velx 초깃값은 0.0이며, set_velx에 의해 설정 가능)
- acc이 None인 경우 _global_accx가 적용됩니다. (_global_accx 초깃값은 0.0이며, set_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- ref가 None인 경우 _g_coord가 적용됩니다. (_g_coord 초깃값은 DR_BASE이며, set_ref_coord 명령에 의해 설정 가능)
- mod가 DR_MV_MOD_REL인 경우 pos_list의 각 pos는 앞 선 pos에 대한 상대좌표로 정의됩니다. (pos_list=[p1, p2, ...,p(n-1), p(n)]로 이루어질 때 p1은 시작점 대비 상대각도, p(n)은 p(n-1) 대비 상대좌표)
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

⚠ 주의

vel_opt=DR_MVS_VEL_CONST 옵션(등속모션)을 선택할 경우 입력된 경유점 간 거리와 속도 조건에 따라 등속모션을 사용할 수 없을 수 있으며, 이 경우에 변속모션 (vel_opt=DR_MVS_VEL_NONE)으로 자동 전환됩니다.

리턴

값	설명
0	성공
음수값	오류

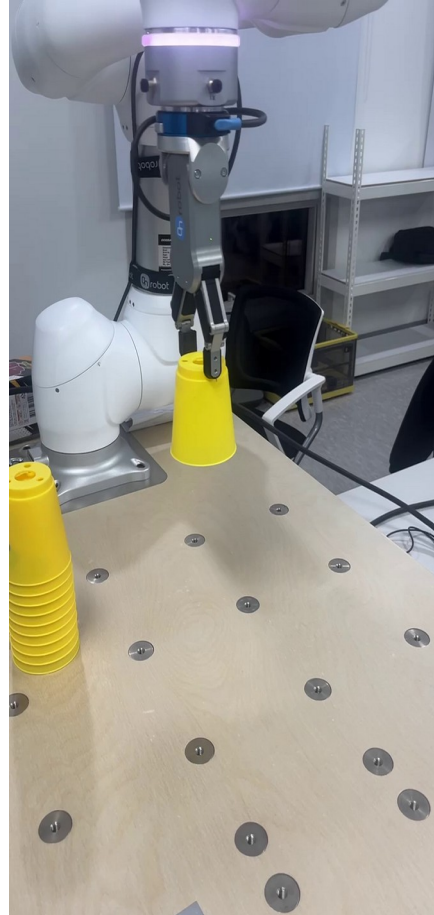
2. 코드 리뷰

4 단 컵 쌓기

```
movej(fourth_ready_j, vel=VELOCITY_J, acc=ACC_J)
movej(fourth_grap_j, vel=VELOCITY_J, acc=ACC_J)
grip()
movej(fourth_middle_j, vel=VELOCITY_J, acc=ACC_J)
fourth_up=fourth_x[:]
fourth_up[2]+=50
movel(posx(fourth_x), vel=VELOCITY, acc=ACC)
release()
```

- 마지막 cup 의 경우 뒤집어야 하기 때문에 movej() 를 활용
- movej() 의 경우 보다 안전한 움직임을 위해서 vel, acc 를 10으로 지정

3. 퍼포먼스



4. 프로젝트를 마치며

배운것들

- # 프로그래밍
- DART-Platform, DART-Studio 가 아닌 VSCode 에서 DSR 패키지를 활용하여 코드를 프로그래밍하는 것을 배움
- 협동로봇을 활용하여 팔레트 물건 정렬하기, 기어조립, 젠가, 컵 쌓기 등 다양한 작업 수행

어려웠던 부분

- # 로봇 제어의 정밀성
- 정밀한 좌표계 계산과 정확한 경로 계획 수립 => 원활한 작업 수행과 충돌 및 오류 방지
- # DSR 함수 사용
- DSR 패키지 안에서 유용한 함수들을 파악하고 사용하는 것
- 함수 parameter 와 return 값을 파악하는 것

느낀점

- # 정확하고 안전한 작업
- 협동 로봇을 어떤 좌표로 어떤 움직임으로 갈건지 확실하게 정하고 정확한 좌표 측정과 안전한 움직임
- # 관련 문서 활용
- 패키지 개발자 (두산로보틱스) 측에서 제공한 문서를 적극적으로 참고하여 사용 방법 (함수 parameter 와 return 값) 을 숙지해야 함